Active Shift Attention Based Object Tracking System

by

Aisha Ajmal

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Science
in Computer Science.

Victoria University of Wellington
2020

Abstract

The human vision system (HVS) collects a huge amount of information and performs a variety of biological mechanisms to select relevant information. Computational models based on these biological mechanisms are used in machine vision to select interesting or salient regions in the images for application in scene analysis, object detection and object tracking.

Different object tracking techniques have been proposed often using complex processing methods. On the other hand, attention-based computational models have shown significant performance advantages in various applications. We hypothesise the integration of a visual attention model with object tracking can be effective in increasing the performance by reducing the detection complexity in challenging environments such as illumination change, occlusion, and camera moving.

The overall objective of this thesis is to develop a visual saliency based object tracker that alternates between targets using a measure of current uncertainty derived from a Kalman filter. This thesis presents the results by showing the effectiveness of the tracker using the mean square error when compared to a tracker without the uncertainty mechanism.

Specific colour spaces can contribute to the identification of salient regions. The investigation is done between the non-uniform red, green and blue (RGB) derived opponencies with the hue, saturation and value (HSV) colour space using video information. The main motivation for this particular comparison is to improve the quality of saliency detection in challenging situations such as lighting changes. Precision-Recall curves are used to compare the colour spaces using pyramidal and non-pyramidal saliency models.

This thesis proposes a motion saliency model where various static and motion features are integrated with the object tracking system to track multiple moving objects. Different features are integrated using pyramidal and non-pyramidal approach and compared by using RGB derived opponencies and the HSV colour space. This provides a new direction in shifting of attention while tracking multiple moving objects. The proposed method is tested with different scenarios and shows that it performs effectively while shifting attention between moving objects.

Dedication

To my parents "Ajmal Saeed Khan" and "Shabana Ajmal".

Acknowledgments

I am very thankful to ALLAH, who give enough strength in completing this work. I am thankful to my beloved parents (Ajmal Saeed Khan and Shabana Ajmal) who always supported me through their constant motivation and asked me to seek higher dreams in life. I am also thankful to my husband *Ibrahim* for giving endless love, support and take caring of kids when I was working.

I am thankful to my supervisors, Dr. Christopher Hollitt (Chris), Dr. Marcus Fearn, and Dr. Harith Al-Sahaf. Dr. Christopher Hollitt has supported me in all these years with his valuable knowledge and experience. I am grateful to him for supporting me in getting Shafi Family Scholarship. I am thankful to Masoor Shafi for awarding this Scholarship. My special thanks is for Dr. Harith Al-Sahaf who motivated me for this work. Without his encouragement and continuous support, it would not be possible to complete this work. He is indeed an inspiration for me. I am thankful to him for dedicating his precious time in reading my work and suggesting the required changes.

I am also grateful to Victoria University of Wellington for supporting me throughout my work by giving all the required resources. I am very thankful to Patricia Stein, who is always so kind and resolved all the administrative issues so we can focus on the studies.

I want to express my love for my four beautiful kids (Maryam, Asma, Hamzah and Yousuf), without their presence and smiles this work was not possible. I would like to thank you to my sisters for their prayers and love.

Finally, I am also thankful to my grandparents for their constant prayers.

Contents

Chapter 1

Introduction

1.1 Domain Overview

Attention allows an agent to select and focus on a part of incoming information for further processing [1]. The human vision system (HVS) has a remarkable capability to use this process for motion perception and scene understanding [2]. The human brain can predict the trajectory of moving objects, like oncoming balls, in one-tenth of a second and split attention across multiple moving objects in the environment [3].

The desire to replicate these basic human vision capabilities in machines is motivated by many applications in scene understanding, traffic flow monitoring, video compression, virtual augmented reality, human-computer interaction like hand gesture recognition, and object tracking [4].

Object tracking is a nontrivial task in which the tracker, along with the estimation of an object's motion has to deal with various issues such as occlusion, object shape deformation, data association and real-time processing requirements [5]. Different approaches have been developed that have used a variety of assumptions to simplify these problems [6, 7, 8, 9]. Examples of such assumptions include assuming smooth object motion with targets having constant velocity or constant acceleration, knowledge

of number and appearance of the objects, and ignoring occlusion [5].

A moving object is a distinct entity having a continuous closed area in an image which needs to be identified by the tracker. Object detection is the process in which an object is localized in the image. On the other hand, object tracking is localizing the object in multiple consecutive frames of a video or image sequences and thereby estimating the path of the moving object. The path of the tracked object (its trajectory) might optionally contain size and appearance information. Each trajectory is unique to a tracked object in a scene [4].

A complete object tracking system is a combination of three important components [5]. The first component is based on detecting the objects present in the video. The second component is the tracker itself which estimates the correct path of the detected objects. The third component is the data association that is used for associating the inter-frame correspondence between multiple objects in the scene.

Different techniques have been previously developed to build trackers, which can be grouped into two general categories [5].

1. **Independent Techniques**: These techniques are suitable in cases having no object interaction or occlusion and all the moving objects, are independent of each other, e.g., trains always uses different tracks to run on and never cross each other.

2. **Dependent Techniques**: These techniques are suitable when the target is affected by other objects or environment. These are used to implement the challenging situations, i.e., self-occlusion, object-to-object occlusion or occlusion with background, e.g., soccer players on a ground, ants in the soil and body cell movements [10].

There are two types of tracking approaches which are as follows [4].

1. **Single Object Tracking**: Several algorithms have been developed to track a single object. These approaches require the design of ap-

pearance models like pose variations or certain motion model, out of plane motion or illumination challenges.

2. **Multiple Object Tracking**: In this tracking category there can be multiple objects in the environment to be tracked. Example of different scenarios to deal with include as a flock of animals like birds or pedestrians on a path. Several issues associated with multiple object tracking make it a challenging research area. For example, occlusions, track initialization or terminations or dealing with objects having a similar appearance such as soccer player wearing the same coloured uniform.

Object tracking can be combined with a visual computational model to replicate the human brain's capability to shift the attention among multiple salient or interesting objects in the environment. These objects have unique attributes that distinguish them from the rest of the image, such as a red ball on the grass.

The attentional shift is a mechanism that increases the information processing about the selected object by inhibiting processing of unnecessary details. This thesis explores this attentional shift mechanism driven by object location uncertainty to propose an active attention-based, object-tracking method.

1.2 Problem Statement

An enormous amount of research has been done to find the optimal sequential location of the moving object by locating its position and correspondence in every frame of the video [5, 11, 12, 9, 13].

With the growth of data volume, the information overload problem is one of the biggest challenge faced by these tracking techniques [11]. Current computing techniques cannot process all information received from the environment using a camera. The data volume increases further when

multiple cameras are involved in tackling real-time problem. Object detection methods have been integrated with visual computational models to limit the intake of information form the images or video. This integration is named as *saliency detection*.

Different trackers, e.g, region, active contour, feature and model-based have been investigated to simulate the human vision system using computational visual attention models to detect salient objects [5, 11, 12, 9, 13]. Informally, a salient object can be considered an interesting object which gets attention due to the uniqueness of some feature relative to its surround. For example, one pink flower among the several white flowers is salient, though it may not be salient if it is among other pink flowers.

The Itti et al. model [14] is one of the prominent saliency-based biologically inspired models that seeks to explain the human visual search by introducing the idea of the shift of attention. This model can be used with a tracker to identify the salient object in each frame of the video. However, if this object disappears or is occluded in any frame, then it can affect the tracking performance. In such situations, an attentional performance mechanism can be useful to gain information about the lost object [15].

Object tracking using attentional shift mechanism is an active research area to tackle the issues that can arise during the integration of the visual attention computational model with the tracking system. The data association in each frame is one of the main challenges that require the labelling of the detected object while tracking. Occulsion is a second issue that must to be tackled for accurate tracking. [7, 16].

This thesis presents an effective object tracking method that use an attentional shift process to detect and track salient objects in a scenario.

1.3 Research Objectives

The overall research objective of this thesis is to develop a multiple object tracker that can shift attention from one target to another while tracking

in a complex environment. The following three research objectives will be answered in the development of visual attention-based object tracker:

1. *To design an effective and efficient visual attention based, multiple objects tracking model.*

   Problem: Many multiple object tracking models have been proposed in the past [4, 5, 9, 17, 18, 19]. These techniques usually use heavy pre-segmentation and complex processing approaches, which makes them inefficient. On the other hand, attention based active vision techniques have shown significant performance boost over conventional models in various static image applications such as object recognition [20, 21], image compression [22, 23, 24] and salient object detection [25, 26]. Some attention-based tracking systems exist that uses fixation of gaze mechanism, to shift attention from one salient object to another [**?**, 27, 28, 12, 13].

   Main task: The main goal of this research objective is to combine a tracker with salient object detection to improve efficiency by shifting attention from one object to another while tracking multiple salient objects. The proposed system will be able to drive attention using its internal uncertainty about the targets' trajectories.

   Sub-tasks:

   - Different simple test scenarios will be developed that simulate the movement of multiple objects using constant velocity or constant acceleration motion models. There will be some scenarios that will include occlusion challenges.

   - The visual attention computational model will be combined with a tracking algorithm that can track multiple objects by using saliency information. The mean square error between the

estimated and true values will be used to quantify the overall effectiveness of the system.

- Comparison with a non-attentional (alternating) model will be performed to analyse the performance of the system. The means square error rates between the estimated and true values of both models will be compared to examine the performance.
- Challenging scenarios, i.e., occlusion handling will be used to test saliency detection and performance of estimation of the location of an occluded target. It be analyse using the mean square error between the estimated and true locations for the occluded object.

2. *To investigate the colour spaces that can help in providing perceivable information for extracting interesting regions form an image.*

   The second objective will provide a detailed colour space analysis by defining the overall image content in term of the different colour spaces such as red, green and blue (RGB) derived opponencies or the HSV colour space.

   Problem: The main challenge while computing the object saliency is to decide which colour space can give perceivable information that can help in extracting interesting areas from an image. Different approaches have combined tracking algorithm with saliency, but have not identified a specific colour space that can contribute to the identification of salient regions effectively [4, 5, 9]. The Itti et al. model [14] has used RGB colour space model for saliency detection, but this is unlikely to be ideal in video data because RGB is not robust colourspace for object detection in the presence of lighting changes. There is a high correlation between the components of this colourspace that mixed the illumination information. It is computationally inefficient to process all components to analyze the same

information three times[29]. Hence, there is a need to investigate colour spaces that can help in determining saliency in a scenario.

Main task: Experiments will be carried out on different datasets to measure the tracking performance while using different colour spaces.

3. *To develop and integrate a motion saliency method with the tracking component of the system.*

   The third objective is to develop a motion saliency method based on visual features and temporal analysis of motion that can be incooperated with the tracking component of the system. This enables the final system to shift attention from one moving object to another based on motion saliency.

   Problem:

   The main challenge while developing the object tracking using the visual attention computation model is to determine the relevant features to acquire new targets [4, 5, 9, 14]. Previous techniques based on parametric motion estimation or block-based matching are computationally intensive [30, 5].

   Some current approaches which are based on pre-segmentation techniques have combined the static multi-features like colour, intensity and orientation with motion features [9]. Only some methods have considered object tracking without segmentation, but it costs additional computation overhead as static multi-features are combined with motion feature [13, 11].

   Hence, there is a need for appropriate selection of static and motion features to combine with the detection process to decrease the overall tracking estimation overhead.

   Main task: The main task is to select relevant static features that can be combined with motion features such as colour, temporal dif-

ferencing, Optical flow and background subtraction, for tracking. Hence, a dynamic feature selection method can be used to select relevant features to increase the efficiency of motion saliency mechanism.

There are three major contributions corresponding to the three goals that are as follows:

1.4 Major Contributions

1. This thesis has shown the integration of object tracking using the Kalman filter and visual computation model that alternates attention from one target to another using a measure of current uncertainty. We present results showing the effectiveness of the tracker in reducing the mean square tracking error. By taking measurements of the location, the tracker using the Kalman filter drives the uncertainty to a low level when it pays attention to an object. The uncertainty of the object grows to a higher level when the tracker shifts attention to another object having high uncertainty from the previous frame. Our proposed tracking approach is tested with different scenarios and shows effective performance in shifting attention to the region that represents a scene.

   The proposed method is able to handle occlusion and predict the location of an occluded target. The results show that the mean square error is high between the estimated and the true locations for the occluded object as the tracker is not paying attention. In future work, the different types of long-term and short-term occlusion will be handled.

   This work has been published in: A. Ajmal and C. Hollitt and M. Frean, "Active shift attention based object tracking system". *in Proceedings of the International Conference on Image and Vision Computing*

New Zealand, IEEE, 2017, pp. 1-5.

2. In this thesis, a comparison is shown between nonuniform RGB derived opponencies with the HSV colour space. The primary motivation about this particular comparison is to improve the quality of saliency detection in challenging situations such as lighting change. Here, Precision-Recall curves are used to compare the colour spaces using bottom-up pyramidal and top-down non-pyramidal saliency models. Our study concludes that if we combine the Saturation-Value or Hue-Value channels, then this improves the detection of salient objects and gives a higher Precision-Recall curve. We have also shown in this thesis that the RGB colour space gives a low precision score as it detects the object using the colour information present in an image.

   This work has been published in: A. Ajmal and C. Hollitt and M. Frean and H. Al-Sahaf, "A Comparison of RGB and HSV colour spaces for visual attention models." in *Proceedings of the International Conference on Image and Vision Computing New Zealand*, IEEE, 2018, pp. 1-6.

3. This thesis has shown the integration of motion saliency using different features, e.g., to detect salient moving objects from video. Several sub-features such as colour, colour change between different frames, optical flow estimation and background subtraction are explored. We have created different challenging scenarios with multiple objects for experimentation. Here, the precision-recall curve is used to compare various features and presenting the effective ones.

1.5 Organization of Thesis

The remainder of this thesis is organised as follows. Chapter 2 presents the background and literature review on the different object tracking and

detection techniques. In Chapter 3, the preliminary tracking system is presented, and the results are discussed. The colour analysis and motion saliency results is given in Chapter 4. Conclusions and future works are presented in Chapter 5.

Chapter 2

Literature Review

This chapter reviews the fundamental concepts of active vision based object detection and tracking. It introduces related background and highlights the important issues of this research area. An important operation called feature extraction that can help identify specific objects from video is discussed later in this chapter. This chapter then reviews previous work that is directly related to the research objectives of this thesis.

2.1 Background

Computer vision is an active research area, and it aims at replicating aspects of the Human Visual Attention (HVA) system. Its application can be seen in different technologies such as surveillance, virtual reality, navigation, traffic control, medical assistance, motion based recognition, video communication and robotics [5]. Different computer vision techniques such as object detection, classification and tracking are used to implement these technologies.

One of the important domain of computer vision is object tracking in a sequence of images. The structure of a simple object tracking system is shown in Figure 2.1. Here, object detection is performed independently for each frame of the video. Different sources of noise can affect detection

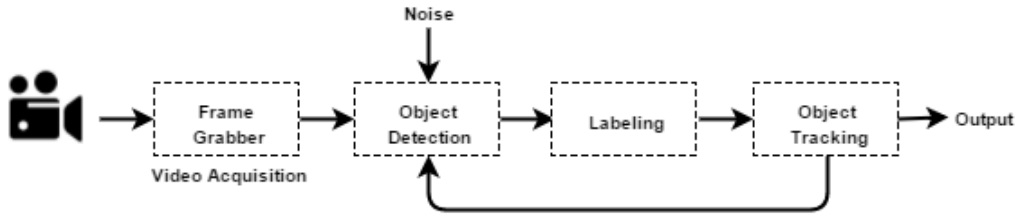


Figure 2.1: A general object tracking system.

results by masking the true location of the object. Tracking is performed after labelling the objects to associate each target with the corresponding target from the previous frames.

Object tracking can be viewed as a state estimation task. A state is a discrete set of value at each time step that contains information about location, target, velocity or acceleration and sometimes additional information such as appearance. The main goal of a state estimator is to select the best estimate of object state from inaccurate, noisy and uncertain measurements. Here, the estimated path that is followed by the moving object is identified as a trajectory.

A straight forward tracking system as in Figure 2.1 does not affect where/how future detection takes place. Trackers can be supplemented by higher-level sensor management modules. One possibility would be to mimic the HVA system using visual computation. Using this computation, interesting called salient region can be extracted from the images (as described in Section 2.6). This extraction of salient information is called saliency [14].

2.2 Recursive Bayesian Estimation

Optimal filtering is a term used for minimizing the error generated from inaccurate measurements while estimating the desired value or state in a time-varying system.

In a dynamic system, there can be uncertainties in the system known as a process noise. Different methods such as smoothing can be used to remove the state inaccuracies by combining the current and previous states of the bayesian estimator [31].

In a nonlinear moving object scenario, the state, $\boldsymbol{x}_t$, depends on $t?1, 2, 3, \ldots$. The dynamic equation for the state change is:

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, t) + \boldsymbol{w}_t, \tag{2.1}$$

Where $\boldsymbol{w}_t$ depends on $t?1, 2, 3, \ldots$ denotes the white noise. The white noise is an independent noise that has uncorrelated samples. The measurement equation is given as follows:

$$z_t = h(\boldsymbol{x}_t, t) + \boldsymbol{v}_t, \tag{2.2}$$

The z_t is a measurement of the target's position, which could be used to estimate the full state. Here, v_t is the independent white noise. During tracking, the state x_t is estimated from the measurements. The state's construction is described by the probability density function $p(x_t|z_{(1,\ldots,t)})$.

The tracking problem can be solved in two steps by any Bayesian filter. The first step is the prediction step that includes dynamic equation and PDF of the state at time $t-1$ is used to compute the prior probability density function PDF of the current state, i.e., $p(\boldsymbol{x}_t|z_{(1,\ldots,t)})$. The second step is the correction that uses the likelihood function $p(z_t|\boldsymbol{x}_t)$ of current measurements to calculate the posterior PDF $p(\boldsymbol{x}_t|z_{(1,\ldots,t)})$.

The solution that involves the above steps is called a *recursive solution* as the previous output of the system is used in the update.

2.3 Kalman Filter

The Kalman filter is a common implementation of a recursive Bayesian filter that assumes a linear model, and requires some statistical assumptions to initialize. These assumptions include the description of the motion

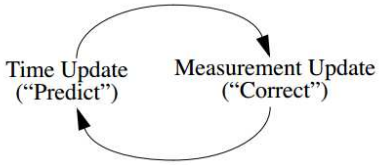


Figure 2.2: Kalman Filter cycle

model, initial location information, initial error covariance of the system, process noise and measurement noise.

The motion model typically depicts linear smooth movement of the target objects in a scene using constant velocity or constant acceleration where velocity and acceleration, remain constant over the time period respectively.

The Kalman Filter provides the optimal estimate of the system using two groups of equations known as the time update equations and measurement update [32]. These equations work together to form a Kalman filter cycle, as shown in Figure 2.2 given in [33]. This cycle is explained as follows:

1. **Time update equation:** The first step of the Kalman filter is the prediction. It is used to obtain the prior estimates for the state at time $t+1$ by projecting the error covariance and the current state forward in time. The time update equation also known as the predictor or process equation defined by linearising the equation 2.1 is as follows:

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{A}}\boldsymbol{x}_{t-1} + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{w}_t \tag{2.3}$$

   where $\hat{\boldsymbol{x}}_t$ is the prior estimate, $\boldsymbol{A}$ is a state transition matrix that is used on the previous state $\boldsymbol{x}_{t-1}$, $\boldsymbol{u}_t$ is a known control input, and $\boldsymbol{w} \sim N(0, Q)$ is a a Gussaian white noise with zero mean and a known covariance $\boldsymbol{Q}$ that captures the deviation from the object motion model in x and y directions. The error covariance matrix $\boldsymbol{P}$ is defined by linearising equation 2.2 is given as follows:

$$\boldsymbol{P}_t = \boldsymbol{A}\boldsymbol{P}_{t-1}\boldsymbol{A}^T + \boldsymbol{Q} \tag{2.4}$$

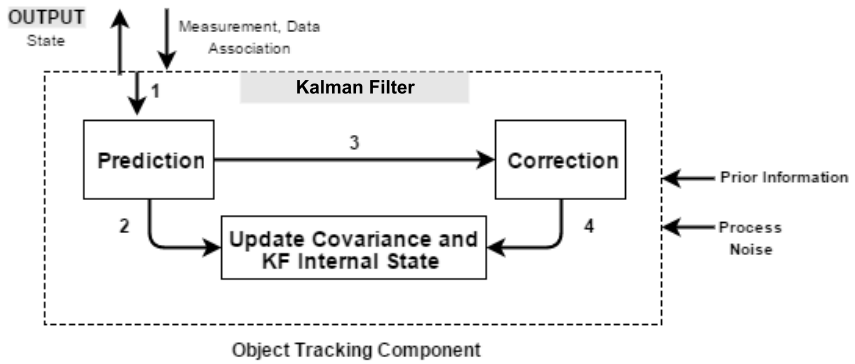


Figure 2.3: Object tracking component using prediction and correction steps of a Kalman Filter. The Kalman Filter's internal state and error covariance are updated by prediction and correction steps.

The error covariance matrix $\boldsymbol{P}$ represent the state estimate covariance that is adjusted over time by the Kalman filter. If this matrix is set too high, then it shows the high uncertainty about the initial measurements. In practice, the process noise covariance $\boldsymbol{Q}$ can be challenging to find empirically.

2. **Measurement update equation:** The correction is the second step that uses the measurement update or corrector equation to correct the estimate by an actual observation at time t.

   The update process begins with the calculation of the Kalman gain $\boldsymbol{K_{t+1}}$ using:

$$\boldsymbol{K}_{t+1} = \boldsymbol{P}_t \boldsymbol{C}^T (\boldsymbol{C} \boldsymbol{P}_t \boldsymbol{C}^T + \boldsymbol{R})^{-1} \tag{2.5}$$

   Here, $\boldsymbol{K}_{t+1}$ is the optimal Kalman gain that minimises the posterior error covariance, $\boldsymbol{C}$ is a measurement matrix, and $\boldsymbol{R}$ is the measurement noise covariance matrix that models the disturbance in the sensors and is a Gaussian white noise sequence with a known covariance $\boldsymbol{R}$ and a zero mean. The measurement noise $\boldsymbol{R}$ is defined as:

$$\boldsymbol{R} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \tag{2.6}$$

Here, σ_{xy} is the covariance between state variables. We have assumed that x and y are independent of each other as there is no correlation when covariance is zero.

$$\hat{\boldsymbol{x}}_{t+1} = \hat{\boldsymbol{x}}_t + \boldsymbol{K}_t(\boldsymbol{z}_t - \boldsymbol{C}\hat{\boldsymbol{x}}_t) \tag{2.7}$$

In this equation $\hat{\boldsymbol{x}}_{t+1}$ is called as posterior estimate at time t. It is a combination of prior estimate $\hat{\boldsymbol{x}}_t$ and using new measurement $\boldsymbol{z}_t$ values. The following equations updates the error covariance.

$$\boldsymbol{P}_{t+1} = (1 - \boldsymbol{K}_t\boldsymbol{C})\boldsymbol{P}_t \tag{2.8}$$

The measurment equation is as follows:

$$\boldsymbol{z} = (\boldsymbol{C}x + \boldsymbol{v}) \tag{2.9}$$

where $\boldsymbol{z}$ is the measurement at time t and $\boldsymbol{C}$ is a measurement matrix and $\boldsymbol{v}$ is the measurement noise with the known covariance.

2.4 Motion Models

The Kalman Filter requires a dynamic model so that it can make predictions. Common assumptions are that the targets move with constant velocity or acceleration. The initial motion models that used to describe the linear smooth movement of the objects are as follows:

2.4.1 Constant velocity motion model

:

The constant velocity motion model depicts the object motion having constant velocity over a time period of tracking. It requires the use of following state vector:

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{x} & \dot{\boldsymbol{x}} & \boldsymbol{y} & \dot{\boldsymbol{y}} \end{bmatrix}^T \tag{2.10}$$

where $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\boldsymbol{y}$, $\dot{\boldsymbol{y}}$ represent the position and velocity in x and y directions, respectively.

The kinematic equation for constant velocity model are as follow:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \dot{\boldsymbol{x}}_{\boldsymbol{t}} dt \tag{2.11}$$

$$\dot{\boldsymbol{x}}_{t+1} = \dot{\boldsymbol{x}}_t + \boldsymbol{w_1} \tag{2.12}$$

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \dot{\boldsymbol{y}}_{\boldsymbol{t}} dt \tag{2.13}$$

$$\dot{\boldsymbol{y}}_{t+1} = \dot{\boldsymbol{y}}_t + \boldsymbol{w_2} \tag{2.14}$$

with

$$\boldsymbol{w_1}, \boldsymbol{w_2} \sim N(0, {\sigma^2}_w) \tag{2.15}$$

The $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ represent the random velocity noise with mean zero and covariance ${\sigma^2}_w$ in x and y direction respectively. The above equations determines the motion of the object moving with the constant velocity where $\boldsymbol{x}$, $\boldsymbol{y}$, $\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{y}}$ represent the postion and velocity information respectively. $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ are assumed independent, so ${\sigma^2}_{w_1,w_2}$ = 0.

The following state transition matrix $\boldsymbol{A}$ can be formed using the above kinematic equations:

$$\boldsymbol{A} = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

Where dt is the sampling time for the system.

It can be assumed that dt =1 as a default value. In the constant velocity model, the control parameter$\boldsymbol{B}$ and $\boldsymbol{u}$ will be zero if there is no known external input given to the system.

The elements in the process noise covariance $\boldsymbol{Q}$ matrix are correlated and specify the amount of uncertainty added to the system due to the effect by random noise. This matrix contains the process noise information of the variances and covariances for the state variables the model. It can

be mapped into the following form:

$$\boldsymbol{Q} = \boldsymbol{G}{\sigma_w}^2\boldsymbol{G}^T \tag{2.17}$$

where $\boldsymbol{G}$ is a matrix selected according to the motion model and σ_w^2 is the variance of random velocity noise in both directions. For the constant velocity model the $\boldsymbol{G}$ is as follows:

$$\boldsymbol{G} = \begin{bmatrix} \frac{dt^2}{2} \\ dt \end{bmatrix} \tag{2.18}$$

The process noise matrix $\boldsymbol{Q}$ is random velocity noise with variance ${\sigma_w}^2$ in both directions is as follows:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{G}{\sigma_w}^2\boldsymbol{G}^T & 0 \\ 0 & \boldsymbol{G}{\sigma_w}^2\boldsymbol{G}^T \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \frac{dt^2}{2} \\ dt \end{bmatrix} \sigma_w^2 \begin{bmatrix} \frac{dt^2}{2} & dt \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \frac{dt^2}{2} \\ dt \end{bmatrix} \sigma_w^2 \begin{bmatrix} \frac{dt^2}{2} & dt \end{bmatrix} \end{bmatrix} \tag{2.19}$$

$$\boldsymbol{Q} = \sigma_w^2 \begin{bmatrix} \frac{dt^4}{4} & \frac{dt^3}{2} & 0 & 0 \\ \frac{dt^3}{2} & dt^2 & 0 & 0 \\ 0 & 0 & \frac{dt^4}{4} & \frac{dt^3}{2} \\ 0 & 0 & \frac{dt^3}{2} & dt^2 \end{bmatrix} \tag{2.20}$$

where σ_w^2 is the variance of the random velocity noise. The measurement matrix $\boldsymbol{C}$ is needed to mapped the state into measurement. Our measurement system measures only the target location, $\boldsymbol{z} = (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, where $\hat{x}$ and $\hat{y}$ are the estimates. We get the following measurement matrix:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.21}$$

The initial error covariance matrix P determines the initial uncertainty in the system. It must be set from the context of a particular experiment.

2.4.2 Constant acceleration motion model:

The model required for tracking object moving with constant acceleration over the time period requires the following state matrix:

$$\boldsymbol{x}_t = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} \end{bmatrix}^T \tag{2.22}$$

where x, $\dot{x}$, $\ddot{x}$ represent the position, velocity and acceleration in the x direction respectively. The same is true for y. In projectile motion, $\ddot{x}$ will be zero, but the acceleration $\ddot{y}$ will be $-9.8m/s^2$.

The kinematic equations for constant acceleration motion model are as follow:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \dot{\boldsymbol{x}}_{\boldsymbol{t}} dt + \frac{1}{2}\ddot{\boldsymbol{x}}_{\boldsymbol{t}} dt^2 \tag{2.23}$$

$$\dot{\boldsymbol{x}}_{t+1} = \dot{\boldsymbol{x}}_{\boldsymbol{t}} + \ddot{\boldsymbol{x}}_{\boldsymbol{t}} dt \tag{2.24}$$

$$\ddot{\boldsymbol{x}}_{t+1} = \ddot{\boldsymbol{x}}_t + \boldsymbol{w_1} \tag{2.25}$$

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \dot{\boldsymbol{y}}_{\boldsymbol{t}} dt + \frac{1}{2}\ddot{\boldsymbol{y}}_{\boldsymbol{t}} dt^2 \tag{2.26}$$

$$\dot{\boldsymbol{y}}_{t+1} = \dot{\boldsymbol{y}}_{\boldsymbol{t}} + \ddot{\boldsymbol{y}}_{\boldsymbol{t}} dt \tag{2.27}$$

$$\ddot{\boldsymbol{y}}_{t+1} = \ddot{\boldsymbol{y}}_t + \boldsymbol{w_2} \tag{2.28}$$

with

$$\boldsymbol{w_1}, \boldsymbol{w_2} \sim N(0, \sigma_w^2) \tag{2.29}$$

The above equations determines the motion of the object under the constant acceleration where $\boldsymbol{x}$, $\boldsymbol{y}$, $\dot{\boldsymbol{x}}$, $\dot{\boldsymbol{y}}$, $\ddot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{y}}$ represent the postion, velocity and acceleration information respectively. The $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ represent the random acceleration noise with mean zero and covariance σ_w^2 in both the x and y directions. The following state transition is formed from the

above equation:

$$\boldsymbol{A} = \begin{bmatrix} 1 & dt & \frac{dt^2}{2} & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & \frac{dt^2}{2} \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

The process noise covariance matrix $\boldsymbol{Q}$ contains the information of the disturbance in the state variables as described in Section 2.4.1. For the constant acceleration model $\boldsymbol{Q}$ also shows the effect of the uncertainty caused by unmodelled noise on the acceleration. The $\boldsymbol{G}$ matrix for the constant acceleration model is as follows:

$$\boldsymbol{G} = \begin{bmatrix} \frac{dt^2}{2} \\ dt \\ 1 \end{bmatrix} \tag{2.31}$$

The process noise matrix $\boldsymbol{Q}$ is random acceleration noise with variance σ_w^2 in both directions is as follows:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{G}\sigma^2\boldsymbol{G}^T & 0 \\ 0 & \boldsymbol{G}\sigma^2\boldsymbol{G}^T \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \frac{dt^2}{2} \\ dt \\ 1 \end{bmatrix} \sigma_w^2 \begin{bmatrix} \frac{dt^2}{2} & dt & 1 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \frac{dt^2}{2} \\ dt \\ 1 \end{bmatrix} \sigma_w^2 \begin{bmatrix} \frac{dt^2}{2} & dt & 1 \end{bmatrix} \end{bmatrix} \tag{2.32}$$

$$\boldsymbol{Q} = \sigma^2 \begin{bmatrix} \frac{dt^4}{4} & \frac{dt^3}{2} & \frac{dt^2}{2} & 0 & 0 & 0 \\ \frac{dt^3}{2} & dt^2 & dt & 0 & 0 & 0 \\ \frac{dt^2}{2} & dt & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{dt^4}{4} & \frac{dt^3}{2} & \frac{dt^2}{2} \\ 0 & 0 & 0 & \frac{dt^3}{2} & dt^2 & dt \\ 0 & 0 & 0 & \frac{dt^2}{2} & dt & 1 \end{bmatrix} \tag{2.33}$$

The measurement matrix $\boldsymbol{C}$ for the constant acceleration motion is as follows:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.34}$$

Again we have assumed measurement of only the target's x and y locations.

Due to its recursive and optimal nature, the Kalman filter has found application in many different tracking tasks. In one approachs, the Kalman filter is used to track points in images containing noise [34]. A Kalman filter is also used to predict an object's position and speed in $\boldsymbol{x}$-$\boldsymbol{z}$ dimensions based on stereo camera-based object tracking [35]. Another approach has used the Kalman filter to compute the priori estimates of vehicle location and velocity after using the scale-invariant feature transform (SIFT) algorithm to extract key points of the objects from video [36].

The demand for accurate tracking of multiple moving objects has made object tracking field an active research field of computer vision [5, 4]. Some of the work in object tracking area has been done using the visual attention computational model to accurately detect the interesting objects from the video and track accordingly [13, 28, 12, 11, 9].

Object tracking using a single camera is computationally difficult as the whole scene cannot be captured at once without shifting camera angles. Different active vision techniques are used for object tracking to improve the computational efficiency of the system by using active control of the camera [13, 15, 37]. These techniques resolved issues such as occlusion and limited camera view. A visual attention computational model can be used with active camera control to search target in a top-down manner [11]. Some visual attention computational models do not require preknowledge about the target and salient object can be tracked using the bottom-up approach [38].

2.5 Visual Attention System

Humans have a remarkable capability to perform complex visual tasks such as scene analysis, object detection and object recognition with precision. Human Visual Attention (HVA) is a sophisticated system that splits attention between multiple information observed from the real world. It can perform motion perception, object recognition and scene understanding simultaneously and can quickly detect a red apple on a tree, a white boat in the sea or a green traffic signal [39]. HVA handles the complexity of the real-world environment by deciding on the action that is required [39].

Selective attention is a mechanism that allows humans to shift the gaze towards the interesting data perceived through the visual input. Motion plays an vital role to attract a viewer's attention as moving objects exhibit more salience due to different feature contrasts such as colour, orientation and intensity. HVA requires minimum processing power to extract these feature contrasts of moving objects and shift attention to salient locations [40].

Developing a computer implementation of the HVA using the primitive selective attention mechanism has been an essential area of research. A visual attention computational model is a partial replication of the HVA system that mimics the functionality of human vision using a mathematical approach. The extraction of interesting information from the incoming sequence of images is called saliency. This computation requires an object detection process which extracts the salient region from the image. The intensity of each pixel in the salient region defines the probability of the saliency in that region [41].

Various visual computational models have developed the notion of a saliency map by integrating different features to select the local saliency in the input [1, 14, 6]. This type of salient object detection which solves the complex tasks is currently an active computer vision research area having

applications in object detection and recognition [26], image compression [22] and object tracking [9, 13, 11].

The two main saliency detection approaches are as follows:

1. **Bottom-up saliency:**

   A bottom-up saliency approach has an HVA based structure that is fast and efficient. It helps in shifting the computer's attention to a region that differs from neighbouring pixels. These regions may have different attributes, i.e., colour, orientation or intensity. Different computational methods are proposed to define the bottom-up saliency mathematically [1, 14]. Itti et al. [14] have identified the most widely used bottom-up saliency computational model for visual saliency.

   This approach detects salient regions that are independent of context information of the scene. [42]. Although bottom-up saliency is a fast and efficient mechanism, it is not suitable for some task-driven applications, such as object recognition and classification, as not every salient region is an area of interest. It detects and tracks the target object in a video efficiently and effectively provided that the target object remains salient throughout the scene.

2. **Top-down saliency:**

   The top-down saliency or voluntary attention approach is suitable for task-driven applications that direct the attention towards the object of interest [43]. If a person is looking for a blue paper on a table, then blue shaded areas should catch the attention more the other areas [11]. Only a few models have fully investigated the top-down approach. Wolfe has developed a model that discussed the human visual search [44]. Itti et al. [14] have done some investigation on top-down saliency by capturing the gist of the scene.

Using a combination of top-down and bottom-up models helps provide the best features to define the salient region [45]. This provides a complete framework for the scenario where it requires to detect the salient locations as well as to locate the object using its contextual information. This type of approach ensures that the viewer will be informed about the area of interest while performing the goal-driven search in the scenario [42]. The VOCUS model [46, 6] uses global saliency formed from a combination of bottom-up and top-down approaches. Different cues from data-driven bottom-up saliency collected along with top-down goal-directed features. The bottom-up part of the system is inspired by the Itti et al. model but has been modified to give better performance by including the weighting of bottom-up features that are learned from the target examples [14].

The top-down saliency is a target-specific mechanism in which target-specific weights used to update the saliency results. VOCUS is one the robust method and first fast system which has utilised a top-down approach, but it requires the knowledge of the target-specific features [46, 6].

2.6 Saliency-based Visual Attention Models

Different saliency-based biologically inspired models have been proposed to introduce the idea of the shift of attention from one salient location to another. The Itti et al. model [14] is one of the prominent works in the field of computational visual attention and is an implementation of saliency and Winner-Take-All network (WTA). The proposed model is based on the architecture of Koch and Ullman [47] and other models. It attempts to explain the human visual search using the bottom-up approach. As shown in Figure 2.4 [14], three channels are used for extracting features, i.e., colour, intensity and orientation. Two features, i.e., red or green and blue or yellow, are used for computing the colour opponency feature. One feature is used for intensity. The orientation is calculated at different an-

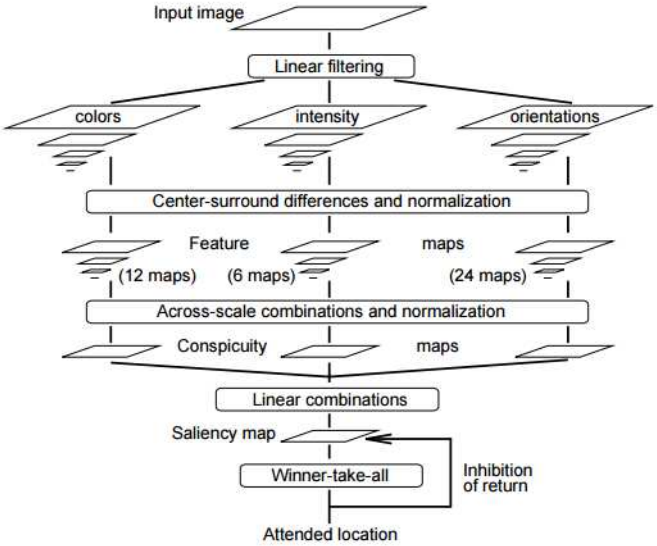


Figure 2.4: Itti's model based on visual attention.

gles 0°,45°,90° and 135° using Gabor filters. The cross-scale difference of Gaussian operation performed a centre-surround mechanism for feature extraction.

Feature maps are used for constructing the conspicuity maps which identify the most salient region. A scalar value is used to represent the saliency during the visual search. The interesting location in the visual area is defined by the help of the spatial distribution of saliency.

The saliency map then followed by the Winner-Take-All network (WTA) and Inhibition of Return (IOR) mechanisms that are used for fixation and gaze shift, respectively. Gaze shift is a process that shifts attention from one object to another. The results from the saliency map are feeds in the WTA mechanism based on a neural network model. The salient regions compete for the highest activation values. The Focus of Attention (FOA) shifts to the location of the winning region. IOR helps the WTA by suppressing the previous salient position and allowing the next most notable location to become the winner subsequently. It prevents FOA for

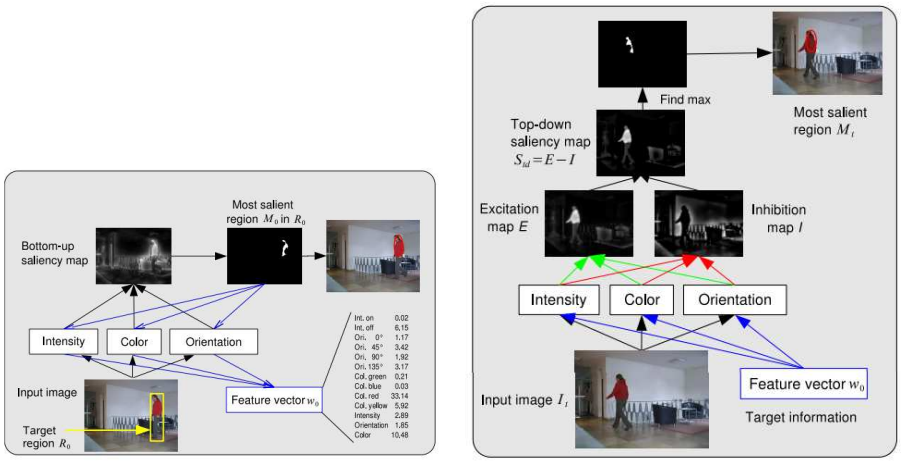


Figure 2.5: Attention based system VOCUS [11] (a) Feature vector are defined by computing the salient region with the user-defined the search area, and (b) Location of a target is determined by computing the top-down saliency map.

immediately jumping to a previously attended site [14].

Although Itti et al. model [14] is an effective implementation of the biological model, different approaches have also proposed to provide more efficient implementations [38, 48, 49, 50]. In one of the approach [49], authors generated the centre-surround effect using Laplacian pyramids and overlapping local patches. These biologically inspired models are beneficial in the implementation of complex processes such as object recognition that can be helpful in the object tracking task by guiding the attention to the salient target.

In [51], the authors developed a tracking method by combining the particle filter and visual saliency mechanism that can handle occlusion and illumination variations by using the colour and saliency distributions. The Bahattacharyya distance is used to measure the similarity between the saliency and colour distributions.

In [6, 11], visual attention computational system VOCUS is developed for tracking object on a mobile platform. Different types of objects ap-

pearance are learned online in real-time. In this system, the user defines a region of interest surrounding it with a drawn a rectangle, as shown in Figure 2.5(a).

There are two different components of their system; one is initialization with three stages. The three phases include the computation of the region of interest using the bottom-up saliency. Then the most salient region is computed using the user-defined area and in last stage describe the feature vector of that region. The best features are extracted using this kind of attention system based on colours, intensity and orientation.

The second component is a object search stage used the top-down approach to perform the visual search to define the most salient region, as shown in Figure 2.5(b). This system does not require a static background while tracking or any particular illumination condition which enables it to work in real-time. The author indicated that the system needs improvement when a target is similar to the background or is close to other objects. Extra features are then required at the expense of additional computational cost.

In another approach, the authors discussed how human's acquired information by focusing selectively when processing a scene while combining information from different fixations over time. This reduces the visual search complexity while focusing on the object of interest and ignoring irrelevant information. The authors used a recurrent neural network to combine the information from different locations to build up the representation of the scene. Instead of processing the whole visual scene, the model selects the locations to attend, thus controlling the amount of computation required.

In [52], authors studied the Kalman filter based on the weighted region matching algorithm, multi-features fusion model and saliency detection. They have concluded that visual saliency computation has shown higher performance than the traditional segmentation, which relies on colour and appearance. The Kalman filter is used to estimate the salient region that

reduces the computational cost.

2.7 Motion Saliency-based Visual Attention Model

A recent interest is seen in the research of the motion saliency to track moving objects from video. Example applications of motion saliency can be considered in the prediction of interesting motion pattern in crowded scenes [53], to track salient object or anomaly detection [54]. Different motion saliency approaches are proposed to identify moving salient objects. For example, a red ball rolling on the floor will have a relative motion difference to the other stationary objects in the scene. Motion saliency determines the state of a moving object using different techniques such as background subtraction, temporal differencing, optical flow and statistical approaches [55, 56, 57, 58].

Temporal information such as motion is one of the essential high-level features that can be incorporated with a visual saliency computational model. HVA shifts more attention to salient moving regions than the salient static region as motion information support different visual tasks such as object recognition, object tracking and scene understanding.

In Itti et al. [59] temporal information is used to proposed motion saliency model based on his previous work [14]. A neurobiological model of visual attention is proposed that has the application to the realistic generation of eye/head movement when watching visual input such as videos. The absolute difference is computed between the luminance of the current and previous frames to incorporate the flicker or temporal variability to the main feature sets. The motion feature is obtained using spatially-shifted differences between Gabor pyramids from the current and previous frames that used the same four local Gabor orientation channels. A diagram of the model is shown in Figure 2.6, where the location

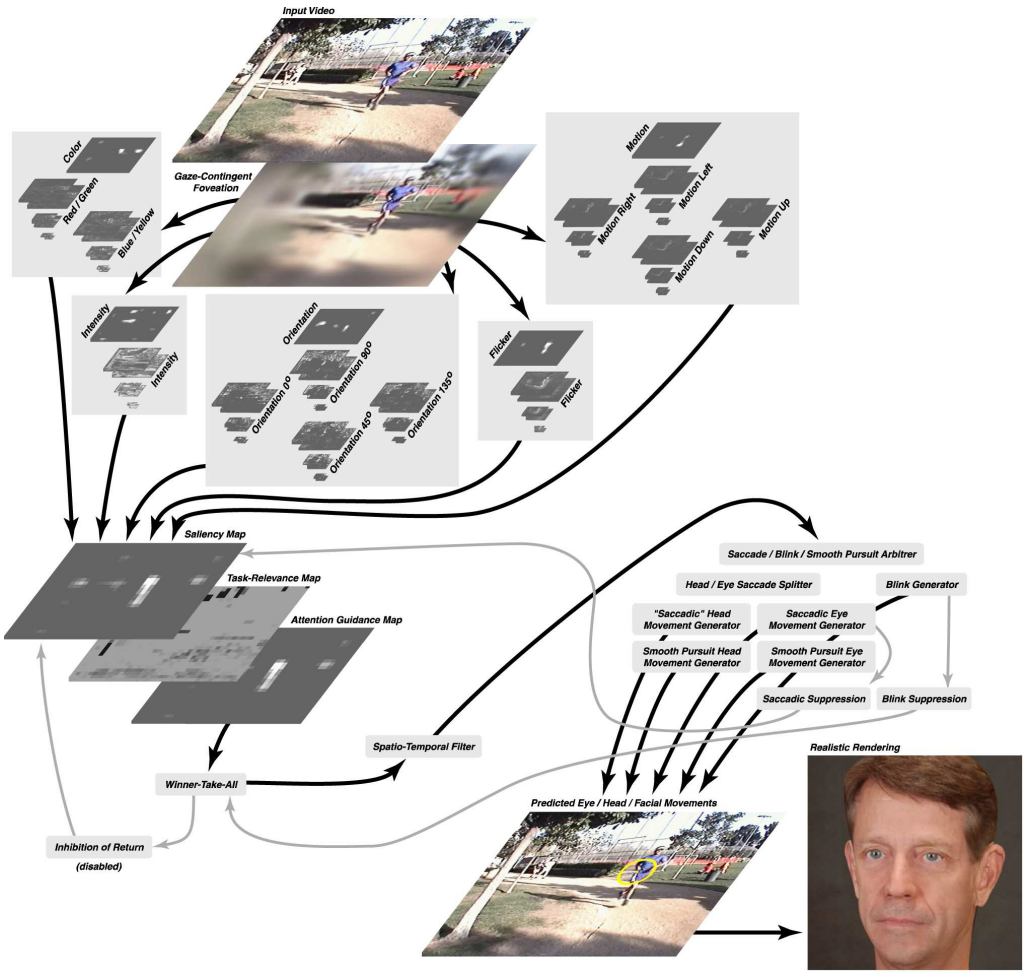


Figure 2.6: Itti et al.'s neurobiological model [59] including motion generated saliency.

with the maximum value gets the attention. The final feature maps yield unique saliency maps in that the maximum location points the salient location. The realistic facial animation is shifted to the resulted covert attention.

There are some limitations of this approach that include the inaccurate detection and labelling the edge of the slow-moving objects as salient, and high computational cost as it cannot be simulated in real-time on a single

CPU. It requires offline motion detection or a computer cluster for real-time processing.

In [9], the motion saliency map is obtained by combining the static multi-feature and motion feature. The saliency detected is done by using seed points for rough segmentation. Then a normalized colour histogram is used for sub-image matching that solves the tracking problem. It is done by describing the salient segmented region to reduce computational cost. The author indicated that this method is invariant to geometric transformations, e.g., and intensity change. The main limitation of this work is the pre-segmentation step that makes it less robust when compared to other algorithms.

Similarly, in [28], the authors developed a mechanism that detects the region of interest according to the motion saliency. Motion saliency maps are generated by exploring the motion consistency using the rank deficiency of 3D greyscale gradient tensors. In addition, the region-based tracking task is done by multiple spatial features of targets that are updated using adaptive templates in each of the frames. The authors indicated that the results of different experiments have shown the proposed method performs object tracking efficiently in the presence of irregular target and camera motion.

In [12], a discriminant centre-surround saliency mechanism is proposed based on a biologically inspired framework in which descriptors for the targets are defined. Using maximum marginal diversity, a subset of features is selected that represents the location of the target in a top-down manner. A motion saliency feature is included in the system by using a simple framework based on a bottom-up approach. Experiments are done to provide three different solutions based on target initialization, feature selection and target detection. The authors indicate that the proposed system is an efficient tracking system when compared to other state-of-the-art trackers.

2.8 Feature extraction and Object tracking

Feature extraction is a significant operation that identifies the essential set of variables to represent a large set of data. During tracking, correct feature extraction plays an essential role in determining the specific object or targets. The operation of feature extraction plays a vital role to discriminate targets depending on the task. Therefore, feature selection should be discriminative and computationally inexpensive.

Different features are used to describe the object in an image. The colour is one of the features that can discriminate objects while tracking because of its low computational complexity and robustness. This feature is successfully implemented to different approaches such as image classification, detection and recognition [37, 60, 30].

Other features such as edge-based are important that describe the image intensities on the boundary using edge pixel differencing. This type of features is insensitive to illumination change and helpful in tracking non-rigid objects [61, 62].

Texture feature can be used to segment and classify regions of interest while tracking object. It provides information about the spatial arrangement of colours or intensity in a frame. Different texture descriptors have been developed, i.e., Grey-Level Co-occurrences Matrix (GLCM's) gives the co-occurrences of intensities in a specific region [63]. Another approach has been proposed in [37] in which colour, texture and motion features are used together to track multiple targets.

There are many other features which are useful in giving cues about the moving objects. Different descriptor like region covariance matrix computes covariance of the region of interest. This can be helpful for object detection and texture classification [64]. These descriptors are also useful in assisting occlusion handling by requiring the occluded features [65]. Depth and gait are some other features that are used to create essential cues while tracking [4].

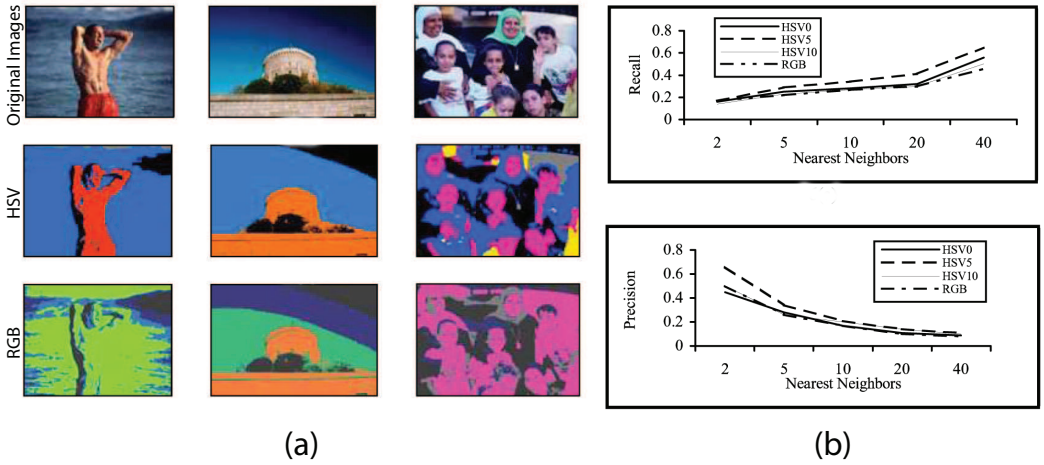


Figure 2.7: The HSV and RGB-based colour spaces are used for segmentation and histogram generation. (a) Comparsion between the HSV and RGB based results, and (b) Precision-Recall curves based on the HSV and RGB histogram [72].

2.9 Colour

The colour is an important feature to detect salient region for image understanding and analysis. Different work has been done to analyse different colour space representations that can yield good object detection results [66, 67, 68, 69, 70]. In most of the detection work, the RGB colour space is mainly used, but there are few drawbacks of using this colour space for object detection applications. In particular RGB it is a non-uniform colour space. The Euclidian distance between colours in the RGB space does not correspond to the colour differences perceived by humans [70]. In addition, the components of the RGB colour space are highly correlated because the change in these components depends on the intensity change. If the intensity change in an image, then all three components will change accordingly [71]. The HSV colour space consists of Hue, Saturation and Value components and is closer to human perception. Hue represents the actual colour, and Saturation describes the purity of the colour and Value

represents the amount of light in colour. The desired hue for a target object can be selected and modified using the specific values of its Saturation and Value. Use of colour space separates the luminance component from the colour information that is helpful in the various object detection applications [69, 73, 5].

In [72], the RGB-based feature fails to determine the variations in colour and intensity and cannot differentiate between two visually separable colours when brightness changes. This led to poor performance as RGB-based clustering is not able to detect object boundaries as high intensities cannot be recognised as shown in Figure 2.7. On the other hand, an HSV-based approach can differentiate the variations between intensity and shade near the boundary of an object. It can retain the colour information of each pixel. Thus they concluded that the HSV-based features are useful in computing segmentation algorithm.

Different approaches have used RGB colour space to detect saliency from the images or video datasets [74, 14, 75, 76, 77, 78]. A few have analysed other colour spaces to detect saliency from static images or video sequences. The different approach shows that the HSV colour space gives high performance in term of visual consistency than the RGB colour space.

One of the work [79], explored different colour representation and proposed a high-dimensional colour transforms. This transformation map uses several representative colour spaces such as RGB, CIE L*a*b* (CIELAB), HSV to combine into a high-dimensional vector. This technique finds the optimal linear combination of colour coefficients in the high-dimensional colour space and linearly separates the salient regions from the background. The performance of this technique degrades of similar colour appears in the foreground and background. The initial colour seed estimation also affects the result and require more feature to be incorporated.

In [80], saliency detection is performed by integrating the local estimation and global search. They have proposed use of a 72-dimensional

feature vector that is computed using RGB, Lab and HSV colour spaces. Another method has used HSV colour space to extract the region of interest from a colour image based on visual saliency [81]. They have combined the colour saliency obtained by using the HSV colour space and Discrete Moment Transform (DMT)-based saliency to get the region of interest. Zhong et al. [82], has proposed a simple method to locate the area of interest from images using the HSV colour space. They found that the saliency map computed using the Value and Saturation components of the HSV colour space get more saliency in the region of interest then background.

2.10 Temporal differencing

Temporal differencing is a simple method to detect moving objects by subtracting the current frame t with the previous frames $t-1$ [83]. The technique involving moving camera usually fails to recognise the whole region of the moving objects or identifies ghost regions (trailing regions) [83]. The trailing areas are detected when the object is moving quickly.

In [84], thresholding is used to select a pixel as a foreground's pixel using a two-frame differencing method. They have combined temporal differencing and image template matching to track and classify targets into human or vehicles. They have concluded that temporal differencing combined with correlation matching allows tracking of the objects in challenging situations such as occlusion without using any predictive temporal filter such as a Kalman filter.

Different approaches have used three frames differencing method to overcome the problems of consecutive frame differencing [85]. One of the issues that two frames differencing method usually creates is ghost regions (trailing areas) instead of detecting the whole moving region when subtracting two consecutive frames.

Another approach [86] segments moving regions using an adaptive

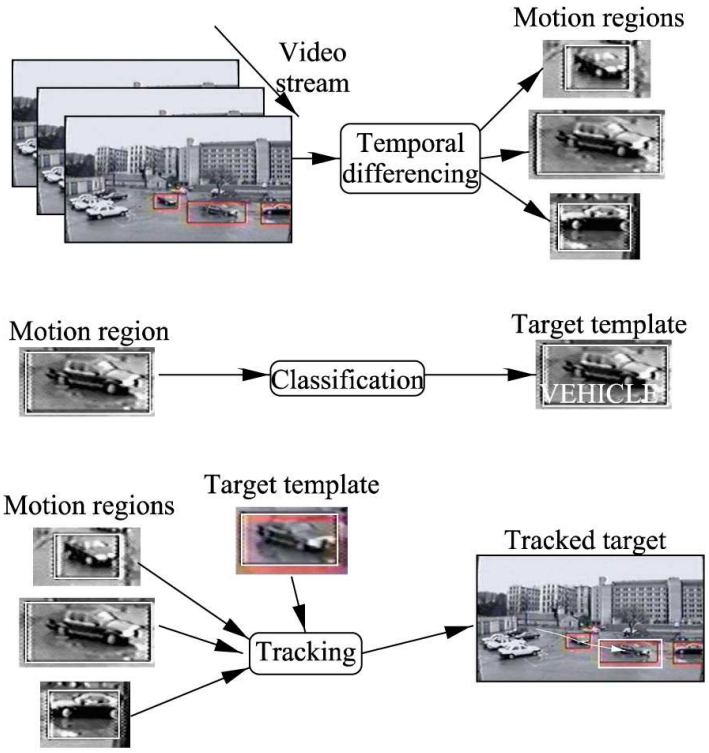


Figure 2.8: Tracking system based on temporal differencing and template matching. Moving objects are detected and classified using temporal differencing and classification metric. Targets are tracked using the motion information and correlation matching. [84].

background subtraction by combining it with three frames differencing method. Here, a three-frame differencing operation determines regions having motion in combination with an adaptive background that detects entire moving region. They discussed the shortcomings of using only an adaptive background without combining it with frame differencing. If the stationary object starts moving, then adaptive background methods leave holes. Frame differencing is not useful in locating the entire shape of a moving object as there is the chance of ghost region (trailing region) detection.

In [59], Itti's motion saliency model is analysed, and linearly, a novel

motion saliency method is proposed using multi-reference frames. These frames are enhanced by spatial saliency information. They discussed that Itti motion saliency model becomes inaccurate when the objects move slowly and in this case objects grouped into the background, and spatially shifted differences makes edge salient instead of the entire object during the computation of feature maps. They proposed a method to overcome this problem by using multi-reference frames and graph-based activation method. Here, graph-based activation method help in differentiating objects from the background using a Markov chain. The proposed method obtains more salient objects and shows higher performance than the Itti motion saliency model.

2.11 Optical flow

Optical flow is a popular method in computer vision to estimate flow of pixel intensity within image sequences. One of the popular methods for determining an optical flow is Horn and Schunck [87] that estimate a motion vector (u, v), where u is the horizontal optical flow and v is the vertical optical flow at every point (x, y) between two consecutive frames $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ at any given time. In this method, it is assumed that images are samples of both space and time such that $\boldsymbol{I}_1(x, y) = \boldsymbol{I}(x, y, t)$ and $\boldsymbol{I_2}(x, y) = \boldsymbol{I}(x, y, t + 1)$, where t is time. They have proposed a brightness constancy constraint that states that a scenario will have the same brightness or no change in the illumination with preserved frame intensities over time such that:

$$\boldsymbol{I}(x + u, y + v, t + 1) = \boldsymbol{I}(x, y, t) \tag{2.35}$$

To solve the two unknowns u, v, Horn and Schunch combined the brightness constancy constraint with global smoothness. Global smoothness constraint defines that is the neighbouring pixel should have smooth optical flow vectors. The energy function needs to be minimised over the flow

fields $u(x, y)$ and $v(x, y)$:

$$\boldsymbol{E_{HS}}(u, v) = \boldsymbol{E_{data}}(u, v) + \lambda \boldsymbol{E_{smoothness}}(u, v) \tag{2.36}$$

where λ is a regularisation parameter that defines the smoothness of the optical flow.

Horn and Schunck algorithm is a global method that solves optical flow estimation based on the global differential methods. Often global differential method results in a dense flow field, but these methods are computationally expensive as many iterations are required to solve optical flow estimation between any images [88].

Another computationally easier method is the Lucas-Kanade method [89]. The Lucas-Kanade method is based on the concept of Least Square method in which the constraint errors are minimised by using a window function $w(x, y)$ that is centred at x, y. It divides the original image into smaller sections and assumes that pixels are moving with constant velocity in that section. Optical flow estimation is performed using a weighted least-square fit for all the pixels in a neighbourhood by considering the constant flow in a local neighbourhood of a pixel under consideration. The energy function equation for Lucas-Kanade optical flow estimation is given as follows:

$$\boldsymbol{E_{LK}}(u, v) = \sum_{x,y} w(u, v)(\boldsymbol{I}(x + u, y + v, t + 1) - \boldsymbol{I}(x, y, t))^2 \tag{2.37}$$

where w is a window function with the assumption of the constraints centred at the centre of the section.

There is another method in which layer-based optical flow estimation is proposed [57, 90] using the Horn and Schunck and Lucas-Kanade algorithms [87, 89, 58] as the baseline method. The layer-wise method of estimating optical flow is by creating a mask indicating the visibility of each layer. The pixel that is located in this mask is used for the matching process. For the layer-based optical flow, three terms are combined such

that data, smoothness and symmetry terms that are as follows:

$$\boldsymbol{E_{data_1}} = \int g\boldsymbol{M}_1(x,y)|(\boldsymbol{I_1}(x+u_1, y+v_1) - \boldsymbol{M}_2\boldsymbol{I_2}(x,y))|, \tag{2.38}$$

where $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are the visible mask of a layer at, respectively, frame $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, u_1, v_1 is a flow from $\boldsymbol{I}_1$ to $\boldsymbol{I}_2$, and u_2, v_2 is a flow from $\boldsymbol{I}_2$ to $\boldsymbol{I}_1$ and The data term matches the two images that are visible in the layer mask. The Gaussian filter is defined as g. The smoothness term is defined as follows:

$$\boldsymbol{E}_{smoothness_1} = \int |\nabla {u_1}^2| + \nabla {v_1}^2|^{\eta}, \tag{2.39}$$

where ∇ is the gradient operator and η varies is chosen between 0.5 and 1. The symmetry matching equation is given as follows:

$$\boldsymbol{E}_{sym_1} = \int |u_1(x,y) + u_2(x+u_1, y+v_1)| + |v_1(x,y) + v_2(x+u_1, y+v_1)|. \tag{2.40}$$

The objective function is obtained by combining the three terms as defined above. It is based on coarse to fine scheme on a dense Gaussian pyramid with image warping. The objective function is given as follows:

$$\boldsymbol{E}(u_1, v_1, u_2, v_2) = \sum_{i=1}^{2} \boldsymbol{E}^i_{data_1} + \alpha \boldsymbol{E}^i_{smoothness_1} + \beta \boldsymbol{E}^i_{sym_1}. \tag{2.41}$$

To optimise this objective function, iterative reweighted least-squares (IRLS) method proposed in [58] is combined with coarse to fine search and image wrapping. For more detail of the layer-based optical flow method, please check [90].

In [91], authors presented a colour based optical flow method and it is compared with the traditional greyscale methods to estimate flow. They have found that linear optical flow methods run faster than greyscale, nonlinear methods. They suggested that the HSV-based optical flow estimation that contains the Value component only, rely on the brightness

information. Whereas the methods based on the Hue and Saturation components of HSV is purely based on colour conservation.

In [53], a saliency detection method was proposed that uses optical flow estimation and performs spectral analysis of image spectra. This method adapted motion spectrum for global motion saliency detection. This detection is considered as interesting regions. The main application of this work is to detect crowd flow.

2.12 Background subtraction

Background subtraction is another method to detect the moving object by subtracting the background image (reference image) from the video sequences. The reference image or the background image can be selected manually or updated using a background model [83]. The reference image should represent the scene with no moving objects.

A common assumption these techniques share is a static background (reference image) with which object of interest is observed. With this assumption, a simple background technique uses a predefined threshold to mark a pixel at location (x, y) as foreground if it satisfied the following:

$$|\boldsymbol{I_t(x, y)} - \boldsymbol{B_t(x, y)}| > \boldsymbol{\theta} \tag{2.42}$$

where $\boldsymbol{I_t}$ is current image, $\boldsymbol{B_t}$ is a background image and $\boldsymbol{\theta}$ is a predefined threshold.

Different methods are categorised in the manner of modeling the Background $\boldsymbol{B_t}$. A simple motion detection technique using background subtraction is to capture or estimate the reference image of a scenario without any moving objects. The estimated reference image can be updated using:

$$\boldsymbol{B_{t+1}} = \alpha \boldsymbol{I_t} + (1 - \alpha)\boldsymbol{B_t} \tag{2.43}$$

where α is a constant in the range $[o, 1]$. If α is 0, then it yields simple background subtraction whereas the value 1 yields frame differencing. Another simple technique used the previous frame I_{t-1} as a reference image

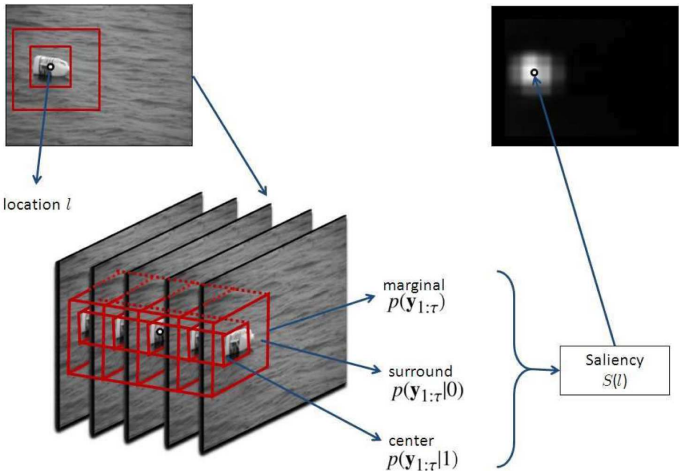


Figure 2.9: Center and surround windows are used for every location l with a collection of spatio-temporal patches extracted from each window. Background points are having lowest centre-surround saliency. All locations that are below to a threshold level a assigned to the background [56].

which reduces to the temporal differencing method. This can be robust to illumination changes, but can detect false regions along with moving objects.

In [56], background subtraction is used to deploy visual attention. They have considered optimal discrimination between the centre and surround stimuli at each location of a visual field as shown in Figure 2.9. Here, background subtraction is incorporated to remove the non-salient locations from the final saliency map. There is no need to update or have a global model for a background. The background is created by considering those locations in a saliency map that are below some threshold by formulating the background subtraction as the complement of the saliency map.

Modelling a background is a challenging task as it requires different situations, i.e., illumination, shadow removal and occlusion, to be considered. Different approaches have investigated background subtraction algorithms to cope with these challenges [55, 92].

Chapter 3

Attention Based Object Tracking

3.1 Introduction

Object tracking is a nontrivial task in which the tracker must estimate the location and motion of a target while dealing with various issues like occlusion, object shape deformation, data association and real-time processing requirements [5].

Different approaches [4] have been developed that rely on a variety of assumptions to simplify the problem. Examples of such assumptions include assuming smooth target object motion such as, constant velocity or constant acceleration motion, knowledge of the number and appearance of the objects, and ignoring occlusion.

Several multiple object trackers have been investigated that simulates human vision system using computational visual attention models. These computational models are often based on the notion of saliency where a salient object is described as a unique entity that should receive increased attention [5, 11, 13, 28, 12]. This salient object must have some outstanding features relative to its surround. For example, while one pink flower among several white flowers is salient, it may not be salient if it is among other pink flowers.

Most of these techniques have addressed the tracking of salient objects

in each frame of the video and assume that the object to be tracked will always remain salient. The effectiveness of these techniques is influenced by whether target loss or occlusion in any frame. In other words, these methods were not developed to tackle tracking of a target in challenging situations such as when lighting might change, or occlusion occurs [5]. Therefore, there is a need to include information about the moving object's state (such as uncertainty) that can be used to drive the shift of attention among different targets.

3.1.1 Chapter Goals

Motivated by the remarkable capability of human vision of motion perception and scene understanding, a new method is proposed in this chapter that integrates a visual saliency computational model with an object tracking algorithm. The proposed method can shift attention from one moving object to another. To achieve this main goal, the following objectives are addressed in this chapter.

- **Generation of Saliency Map**

  Detecting a salient object is an essential part of determining the interesting objects in a frame. Here, moving salient objects are identified using the pyramidal computational approach of Itti et al. [14]. Instead of using all Itti's features, initially, a colour feature is used to detect specific coloured targets for demonstration purposes. In future, can be used with a broader set of features that can have a direct impact for detecting motion from the scenarios.

- **Integration of saliency and tracker**

  The second objective of this chapter is the integration of saliency detection with a tracker. The information of the interesting region values is considered as salient location consisting of targets. To achieve

this objective, the centroidal information of any detected salient objects is computed and used as an input to the tracking system to estimate the future location of the objects.

- **Maintenance of the tracks during tracking**

  Associating the information of the detection with tracking in each consecutive frame is another crucial task during object tracking. Here, Mahalanobis distance [93] has been used to measure the consistency of object detection with expected target locations. The detection in each fame is linked with tracking information using this distance.

- **To construct the attention shift based tracker**

  A new attention shift based tracker is proposed that uses the saliency map and object's state uncertainty information to direct attention from one object to another. This kind of tracker has a direct impact on tackling complicated situation such as occlusion, illumination changes and a moving camera.

3.2 The Proposed Model

This section explains the proposed system in which the visual attention computational model is integrated with the tracking algorithm to track and shift attention to salient targets. The proposed model, referred to as *attention shift tracking* (AST) is shown in Figure 3.1. The system has four modules, detection, association, tracking and attention shifting. These modules each contribute to the estimation of the trajectory of the moving objects across a scenario.

The following steps briefly describe how the tracker shifts attention when integrated with saliency information in the proposed model.

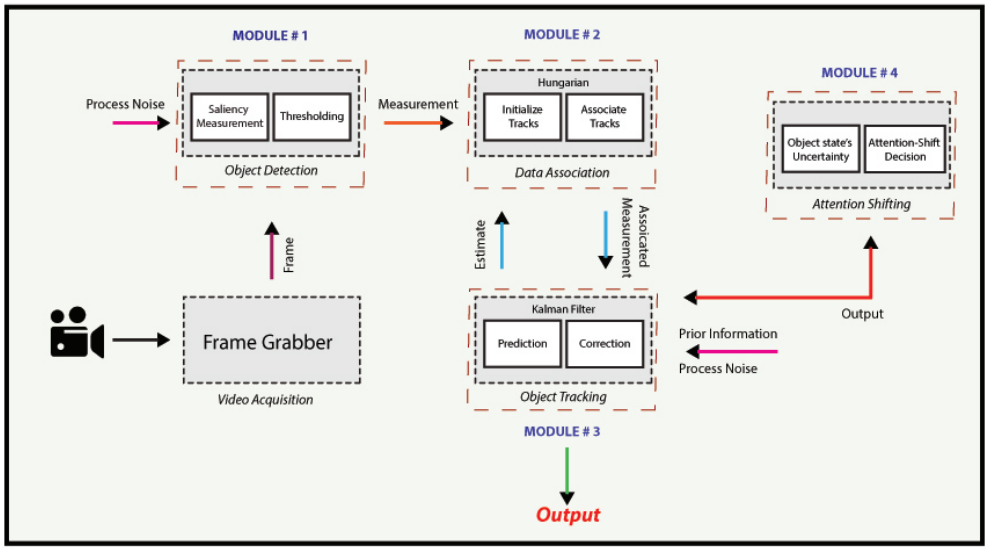


Figure 3.1: The proposed attention shift tracking (AST) model.

1. Extract the frames from the video using a simple frame grabber.

2. Compute the saliency map using the colour feature of the Itti et al. [14] computation model (more details in section 3.2.1).

3. Detect salient objects and compute centroidal information.

4. Feed centroidal information into the tracking algorithm to create the track of the detected objects (refer to section 3.2.3 for details).

5. Associate the estimated tracks with detection to maintain the tracking information (as discussed in section 3.2.2).

6. Drive the attention of the tracker from one object to another using the object's state uncertainty information (refer to section 3.2.4 for details).

3.2.1 Module 1: Object detection

In the first module, labelled as 'Object detection' in Figure 3.1, the computation of saliency takes place. In the proposed model, saliency determination has been completed using a visual attention computation model based on Itti et al. [14] (refer to Section 2.6 on page 24 for details). The traditional Itti model uses low-level features such as colour, intensity and orientation.

We have used colour as the main feature to extract saliency, as shown in Figure 3.2. The Gaussian filter is applied to this feature to generate Feature Maps using center-surround mechanism. The conspicuity map is computed by promoting the maps with the strong peaks. The final saliency map is generated on the basis on this conspicuity map.

In this work, the saliency map is followed by the Otsu's thresholding method to enhance the contrast of the saliency maps. This method chooses the threshold based on the observed distribution of pixels values [94]. First, the image binarized, and then blob analysis is performed to find the centroid locations of the connected regions of the salient objects in each frame.

In [14], a Winner-Take-All (WTA) mechanism is used to select the most salient location in the frame for fixation. The WTA mechanism is not used in the proposed model in this chapter as multiple objects can be present in a scenario which requires a technique to detect multiple centroids for tracking. Itti's Inhibition of return mechanism is also not needed as fixation of gaze is tied by the uncertainty of the location of the object, as explained in Section 3.2.4.

3.2.2 Module 2: Data Association

Data association is a necessary component that links detections from input images and the known target during tracking. Different approaches can be used to solve the correspondence problem between different data.

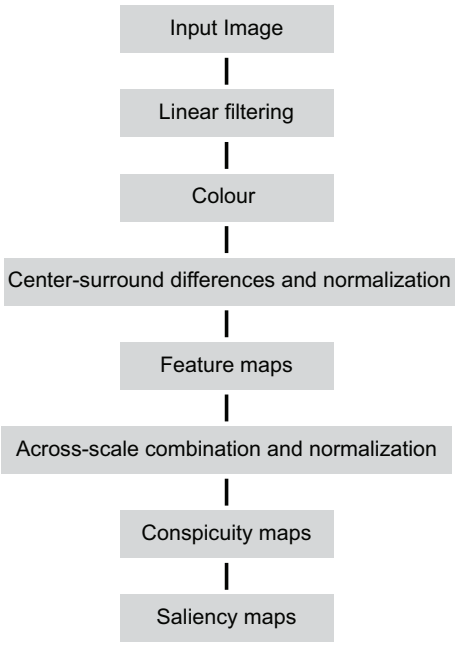


Figure 3.2: Itti's model based on visual attention using only colour feature (redness) [14].

One prominent method is the Hungarian algorithm [95] that provides the solution to the data association problem by minimizing the cummulative cost between detected and estimated positions. In the preliminary system, Mahalanobis distance has been used to measure the distance between the predicted and the detected location of the object. The advantage of using the Mahalanobis distance is that it considers uncertainty. The Mahalanobis distance is computed as.

$$\boldsymbol{D_M} = (\boldsymbol{x} - \hat{\boldsymbol{x}})^T \boldsymbol{P}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{3.1}$$

where $\boldsymbol{x}$ is the detected location, $\hat{\boldsymbol{x}}$ is the predicted location and $\boldsymbol{P}$ is the error covariance matrix.

The Hungarian algorithm is used if there are multiple target detections from a single frame. It uses the sum of the Mahalanobis distance for each track as the cost and assigns the detected objects to minimize the overall cost.

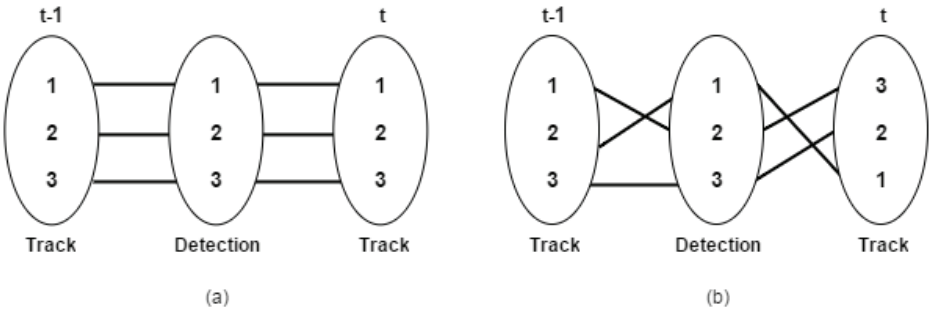


Figure 3.3: Track maintenance (a) Tracks remain the same in each time step for every detection. (b) Tracks changes for the detections during tracking.

In this model, when there are new detections, then new targets are defined as shown in Figure 3.3. The new targets are represented by a unique number called track or detection ID. In Figure 3.3(a), the number of detections in a scene is shown that corresponds to the respective tracks at times $t-1$ and t. If there is an occlusion among different objects then tracks IDs are assigned again to each detection for the track maintenance as shown in Figure 3.3(b). The detection 1 corresponds to track 2, detection 2 corresponds to track 1 and detection 3 corresponds to track 3 at time $t-1$. At time t the tracks corresponding to the detections are updated and maintained the detection and tracked information of the same object over the time.

3.2.3 Module 3: Object Tracking

The purpose of this module is to estimate the trajectory of the moving objects. Here, the predict-update framework of the Kalman Filter used as the underlying tracking algorithm. The following sections describe the configuration of the Kalman Filter's used to develop the tracking system.

3.2.3.1 Initial error covariance matrix

The initial error covariance matrix $\boldsymbol{P}$ determines the initial uncertainty in the *system* when a new target is detected. We do not have any idea about the initial state, and there is no correlation between the values. Hence, we have low confidence that the initial state is close to correct values so. The simulated frame size is 500 $\times$ 500 pixels. Here, following the initial error covariance matrix is assumed for the constant velocity motion model:

$$\boldsymbol{P} = \begin{bmatrix} 200^2 & 0 & 0 & 0 \\ 0 & 200^2 & 0 & 0 \\ 0 & 0 & 200^2 & 0 \\ 0 & 0 & 0 & 200^2 \end{bmatrix} \tag{3.2}$$

Here, 200 is variance computed using the size of the input data in pixels. The initial error covariance matrix used for constant acceleration motion model is:

$$\boldsymbol{P} = \begin{bmatrix} 200^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 200^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 200^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 200^2 \end{bmatrix} \tag{3.3}$$

3.2.3.2 Measurement Noise Matrix *R*

The measurement noise covariance matrix $\boldsymbol{R}$ models the uncertainty in the *sensors* as defined in Section 2.3 on page 13. This $\boldsymbol{R}$ matrix format is used for both of the constant velocity and constant acceleration motion model. We have assumed that motion in the $\boldsymbol{x}$ and $\boldsymbol{y}$ directions are independent, so the off diagonal covariance will be zero.

For calculating the measurement noise of the system, a simulated dataset is used with circular object with radius 25. The saliency of this object is calculated then centroid is found out by thresholding. The error

covariance matrix $\boldsymbol{R}$ is calculated by subtracting these centroids from the ground truth and summing the result. Finally, the Gaussian is fitted to calculate the variance, that is the error distribution of the measurement, as shown in Figure 3.4. The computed error covariance matric as follows:

$$\boldsymbol{R} = \begin{bmatrix} 0.3790 & 0 \\ 0 & 0.3840 \end{bmatrix}$$

.

3.2.4 Module 4: Attention Shifting

In the proposed method, shifting attention is based on the relative state uncertainty of each object. When there are multiple objects in a scene, attention shifts to the object that is most uncertain. The uncertainty is characterised as the area of the uncertainty ellipsoid described by $\boldsymbol{P}$ as shown in Equation (2.4) in Section 2.3 on page 13.

When the tracker is paying attention to an object, it will keep collecting measurements of that object's location and update its state estimate accordingly. When it is not paying attention, then the uncertainty of the state of that object will grow according to the Kalman filter prediction Equation (2.4). If the tracker is required to keep predicting for a long time, then the uncertainty in target location might grow so large that the tracker effectively loses the target. When there are multiple objects in a scene, then the attention will be shifted to the most uncertain object. There are different methods to characterise the overall uncertainty in the object's location. The first possibility position variances summed together, and the tracker shifts the attention to the object having higher variance. The second possibility also gives the same result when the uncertainty is measured by finding the area of the ellipse associated with the state error covariance using Equation (2.8) in Section 2.3 (on page 13).

We can use either method to find the area of the ellipse to alternate the tracker attention because the object with the highest area value will get the

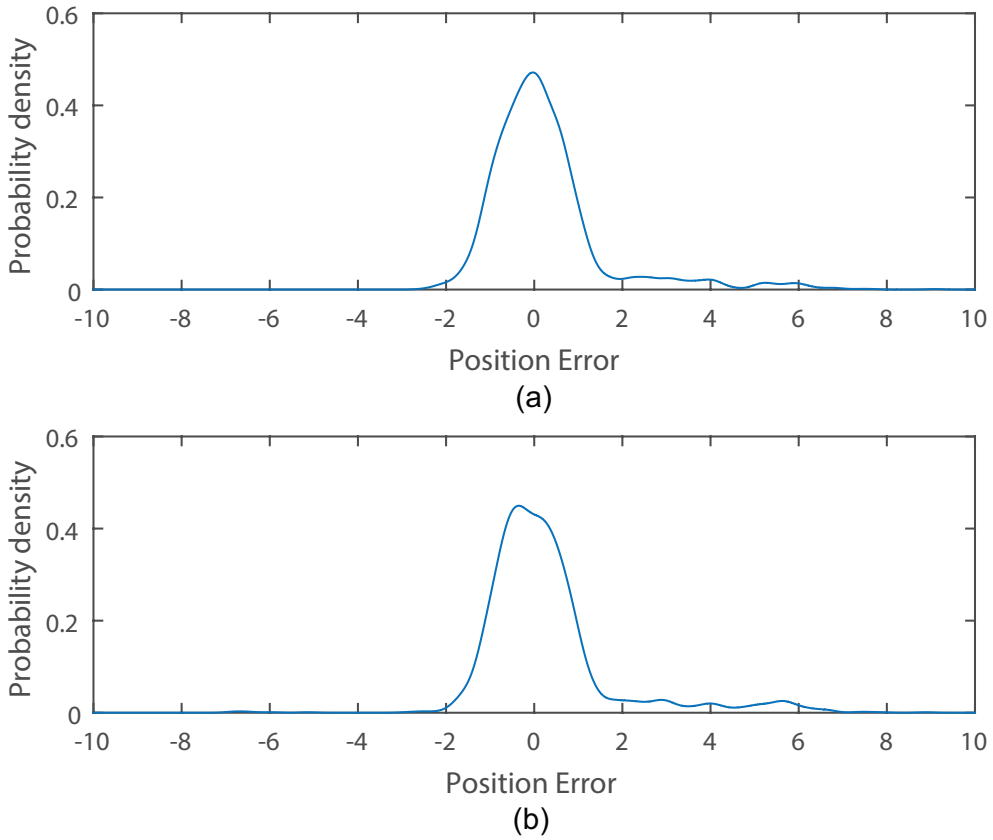


Figure 3.4: Error distribution of the measurement noise (a) Error distribution in x direction, and (b) Error distribution in y direction.

attention of the tracker. In contrast, the object having the lowest area value will get less or no attention.

In the given system, process noise indicates the deviation from the object motion model in x and y directions. While shifting attention, process noise plays an important role. If the process noise is the same for each of the objects present in the scenario, then we would expect all the objects get the attention sequentially. When each object has different process noise, then the object having the highest process noise should get more attention. As it is hard to make predictions about the location of an object with high process noise, so the uncertainty for that object grow more

quickly. In this situation, Kalman filter should have more confidence in the measurements.

Table 3.1: Summary of Datasets used in the proposed method. These datasets are illustrated in Figure 3.5 and generated version is shown in Figure 3.6

Scenarios	Number of Objects	Movement	Main Challenge
1	1	straight/horizontal	none
2	3	straight/horizontal	none
3	1	projectile	none
4	2	projectile	none
5	2	projectile	occlusion
6	4	straight/horizontal/in angle	none
7	3	straight/in angle	occlusion
8	2	in angle	occlusion

3.3 Experiment Design

The primary purpose of the proposed tracking system is to update the uncertainty information of the one target from a saliency map and keep track of the salient targets. To evaluate the effectiveness of this system, simulated datasets are used with different scenarios. The detail of these scenarios is given in the this section.

3.3.1 Simulation Datasets

Different scenarios depict the multiple object movements, as shown in Figure 3.6 were generated as simulated datasets, as shown in Figure 3.5. This testing is done with simulated datasets to ensure that the suggested plan applies to complex real-time situations.

These datasets are summarised in Table 3.1. The frames of 500×500 pixels generated where circular shapes with radius 25 are drawn for each

target. We have used random initial values for the object's location in the first frame and then increment the position of this object in the x and y directions throughout the scenario. A constant velocity (or a constant acceleration motion model used with zero acceleration) to model these movements.

There are mainly two types of scenarios base on occlusion. In Figure 3.5, the movement of the objects in each scene is illustrated. It can be seen that in scenarios 1 and 2 objects move horizontally with constant velocity without any occlusion. The state vector for the constant velocity model is defined using the Equation (2.10) in Section 2.4.1 (on page 16):

The process noise $\boldsymbol{Q}$ used for generating the senario model with constant velocity is as follows:

$$\boldsymbol{Q} = 0.2 \begin{bmatrix} dt^4/4 & dt^3/2 & 0 & 0 \\ dt^3/2 & dt^2 & 0 & 0 \\ 0 & 0 & dt^4/4 & dt^3/2 \\ 0 & 0 & dt^3/2 & dt^2 \end{bmatrix} \tag{3.4}$$

where 0.2 is the variance of the random velocity noise assumed in both direction.

Projectile movement is shown in scenarios three, four and five. In these scenarios, the illustrated movement shows that objects are thrown upward at an angle while moving horizontally with constant speed. After reaching a height, the object falls following the symmetric path with which it moved upward. The vertical acceleration with which the object will move in a projectile is an acceleration of gravity which is $-9.8m/s^2$. The state matrix for the constant acceleration model used in these scenarios is defined in Equation (2.22) in Section 2.4.2 on page 19.

The process noise $\boldsymbol{Q}$ used for generating the senario model with con-

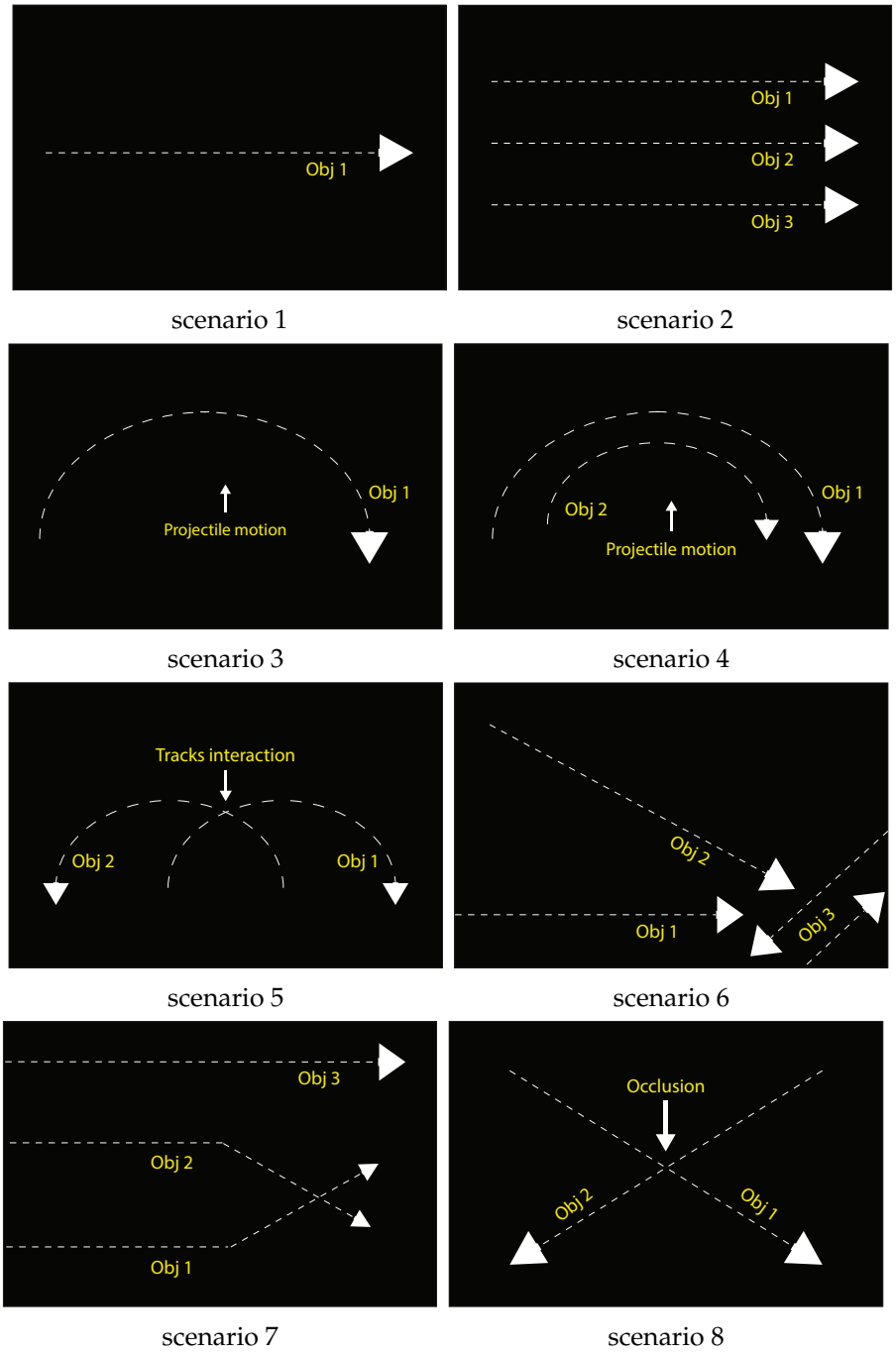


Figure 3.5: Different scenarios have been created for simulation purpose. From left to right and top to bottom, scenario 1 and 2 shows objects moving horizontally, scenario 3 and 4 show projectile motion of single and multiple objects, scenario 5 shows the occlusion situation, scenarios 6, 7 and 8 show random movement of the objects.

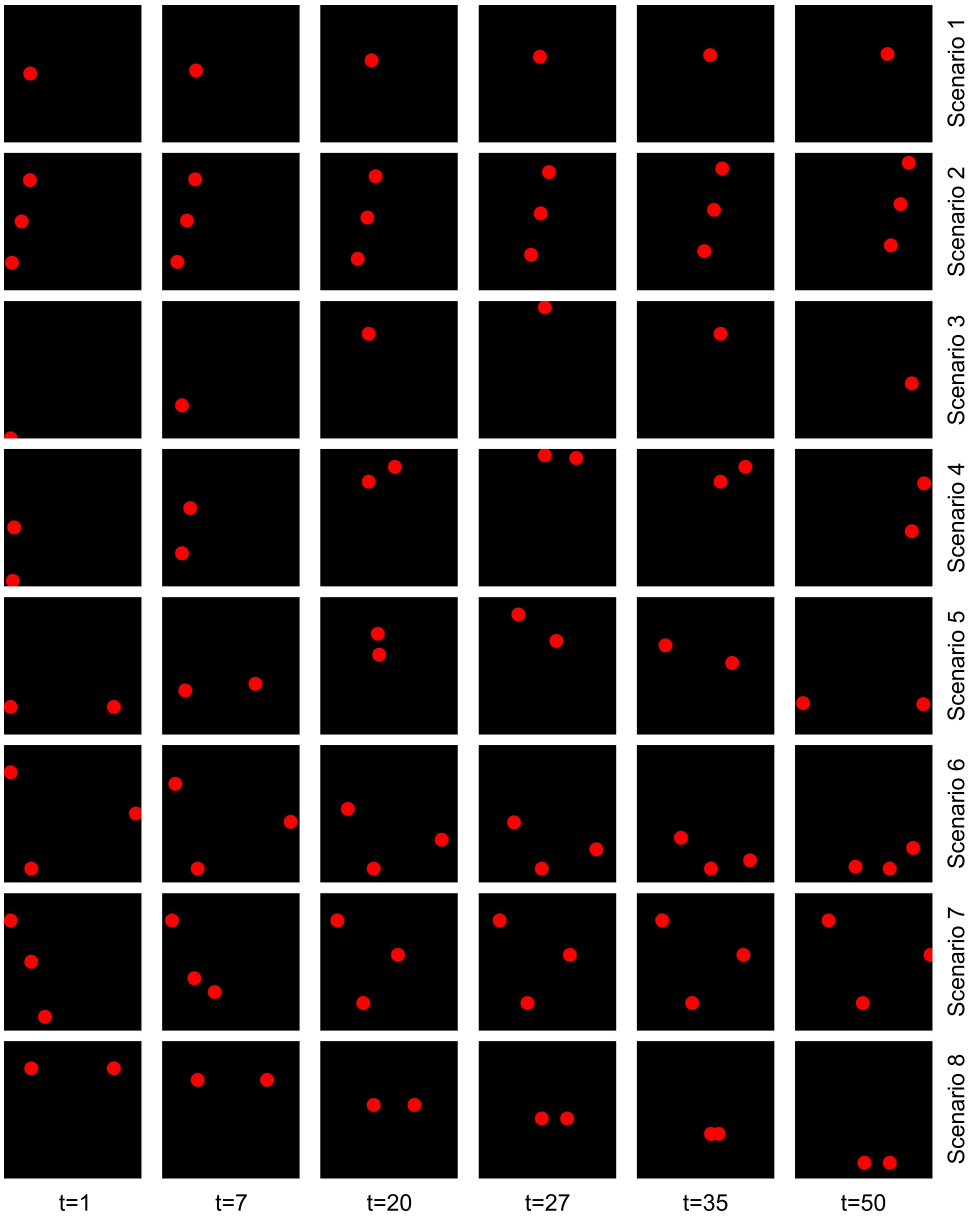


Figure 3.6: From top to bottom, frames at time t is shown from the scenarios depicting movements illustrated in Figure 3.5.

stant acceleration is as follows:

$$\boldsymbol{Q} = 0.2 \begin{bmatrix} dt^4/4 & dt^3/2 & dt^2/2 & 0 & 0 & 0 \\ dt^3/2 & dt^2 & dt & 0 & 0 & 0 \\ dt^2/2 & dt & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & dt^4/4 & dt^3/2 & dt^2/2 \\ 0 & 0 & 0 & dt^3/2 & dt^2 & dt \\ 0 & 0 & 0 & dt^2/2 & dt & 1 \end{bmatrix} \tag{3.5}$$

where 0.2 is the variance of the random acceleration noise assumed in both directions.

3.3.2 Challenging Situations

The scenarios 6, 7 and 8 represent some challenging situations in that some targets come close to each other and occlusion occur. The saliency information of occluded targets combined due to which tracking estimation becomes difficult as shown in Figure 3.6.

In the proposed method, we have used the Kalman filter to predict the location of the occluded target's location and solve this issue. The uncertainty of the occluded targets increases as the tracker is not paying any attention to these targets.

3.4 Results

The result section includes object detection and tracking result. The result of the simulated datasets are shown in Figure 3.7 and 3.8. Here, it is discussed how the saliency result can affect the tracker's shift of attention.

3.4.1 Object Detection Results

In the proposed system, object detection is performed using a visual attention computation model, as discussed in Section 2.6 (on page 24). After measuring the saliency, thresholding is done for finding the centroids of the objects. The saliency detection for different situations can be seen in Figure 3.7. After saliency detection, we have manually annotated the salient objects in the frame to construct ground truth. In scenario 8 the saliency of different objects combined such as in frame 35, as shown in Figure 3.7 and cause the occlusion. Due to occlusion, the saliency information of one of the object is not detected. This condition of occlusion is solved by the Kalman filter which is used as a tracking algorithm by predicting the location of the objects from the previous information when different saliency measurements are combined or not found in some frames.

3.4.2 Object Tracking Results

The proposed model, as shown in Figure 3.1 is adopted to estimate the motion of the object by detection, association, attention shifting and tracking modules. First, object detection is performed to find the saliency information on the dataset then estimated response is computed using the Kalman filter as discussed in Section 3.2.

The main objective of response estimation is to decide which target's location needs to be updating. In the proposed method, we update only one target at a time or in other words, the tracker keeps its attention on one object entirely. The tracker does not pay attention to the remaining targets. If there is only one target in a dataset, then the tracker pays attention to this object only and updates its uncertainty information. If there are multiple targets in a scenario, then tracker updates the location of the most uncertain target as discussed in Section 3.2.4 (on page 49).

The estimated response is shown in Figure 3.8 of the tracking component which is summarised in Table 3.2. The grey label "prediction" repre-

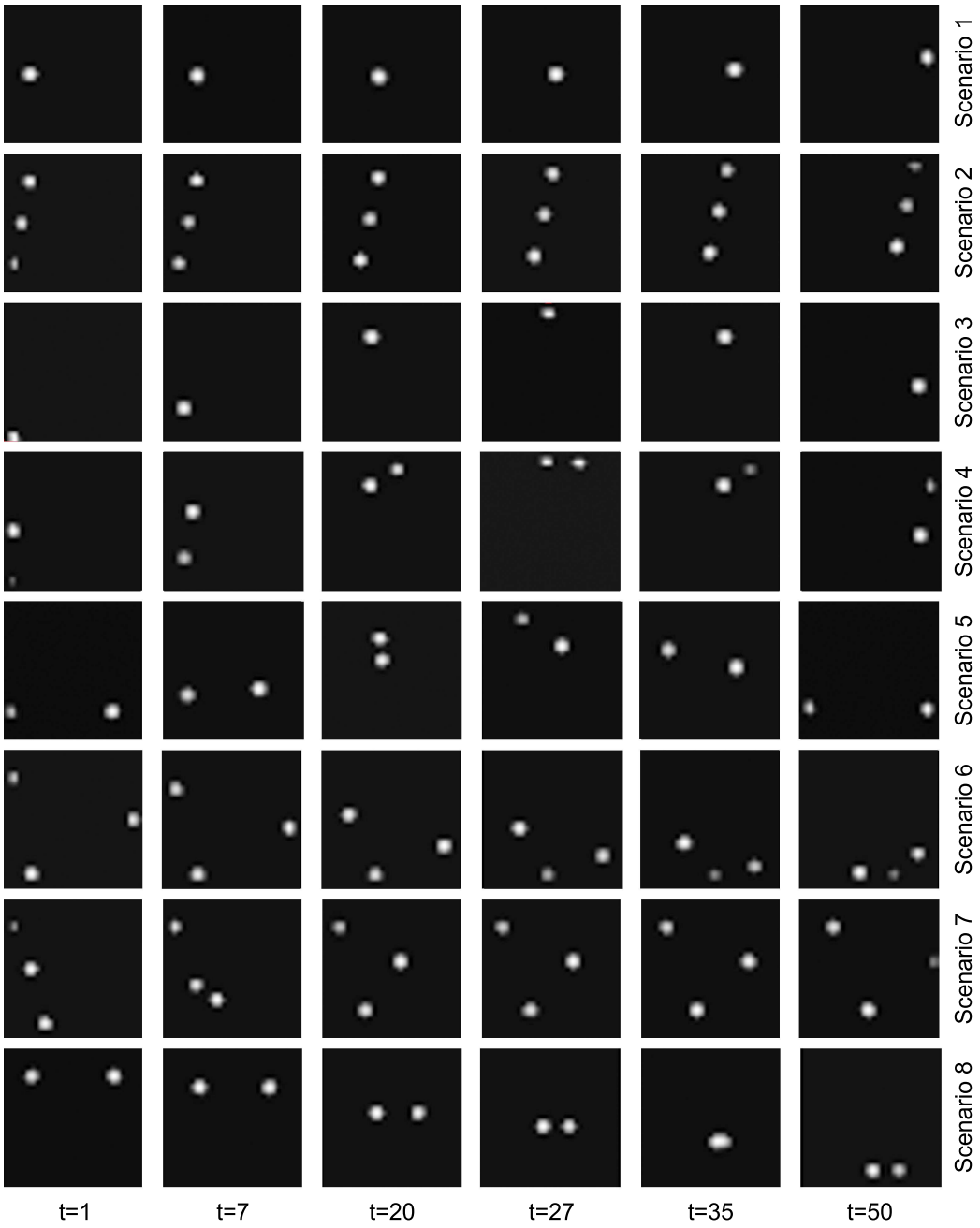


Figure 3.7: Saliency detection of the scenarios shown in Figure 3.6.

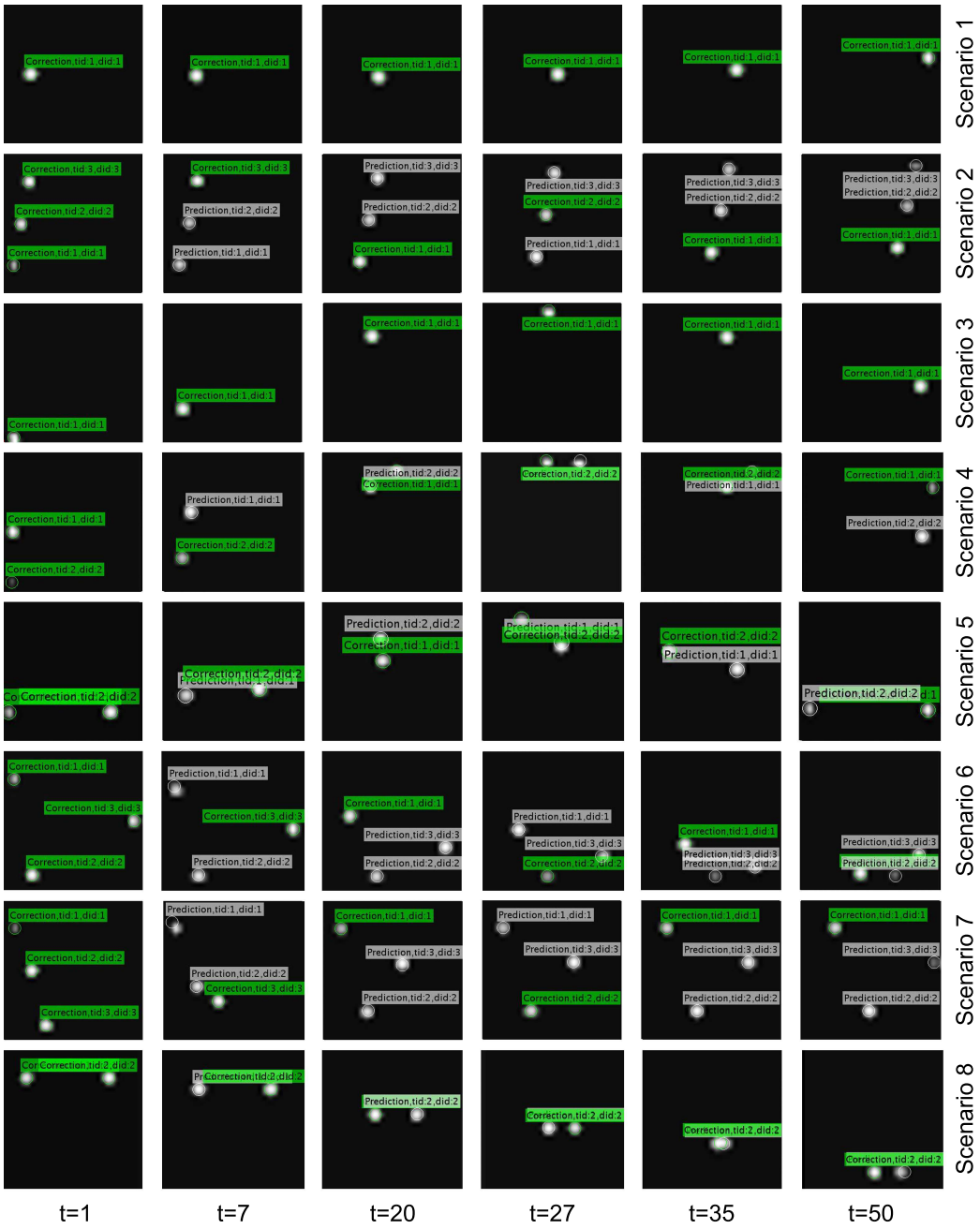


Figure 3.8: Estimation response of the scenarios shown in Figure 3.6.

Table 3.2: Summary of object tracking result as shown in Figure 3.8.

Scenarios	Object	Estimated response	Attention Shift	Situation
1	1	correction	none	none
2	3	correction/prediction	yes	none
3	1	correction	none	none
4	2	correction/prediction	yes	none
5	2	correction/prediction	yes	none
6	3	correction/prediction	yes	none
7	3	correction/prediction	yes	occlusion
8	2	correction/prediction	yes	occlusion

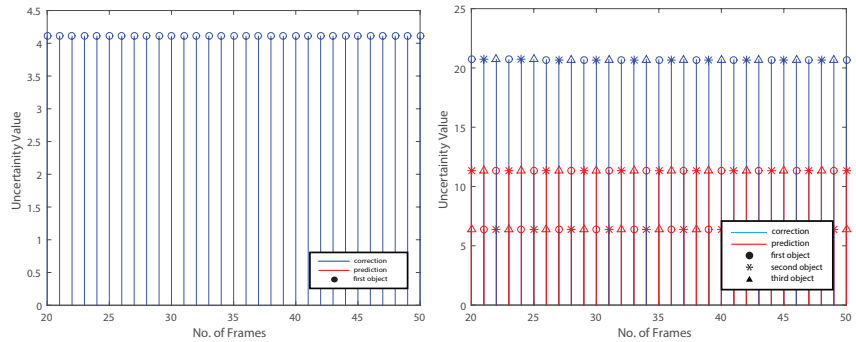
sents that the tracker predicts the object location in the current frame and the green label "correction" shows that the tracker has access to measurement in estimating the object location. The detection ID is the rank number given by the detection component when any object is detected. The tracking ID is the rank number provided by the tracker to the corresponding detection.

3.5 Discussion

The discussion session includes the details about the uncertainty computation and how it affects the shift of focus between multiple targets. Here, eight simulated scenarios discussed in Section 3.3.1 are used to explain the alternation of attention. The tracker updates the most uncertain object and then shifts focus to the next most uncertain target.

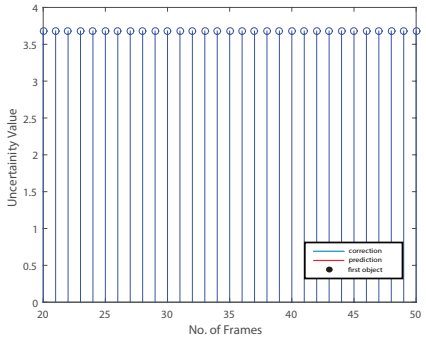
3.5.1 Uncertainty Shift of Attention

The tracker shift of attention is shown in the Figure 3.9. Here, the tracker's estimation response is represented by different colours. The red colour represents the prediction, whereas the blue colour denotes the correction. The shapes represent the number of targets.

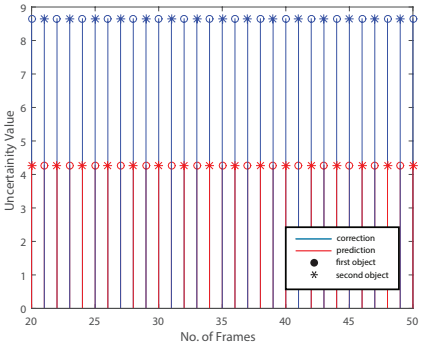


scenario 1

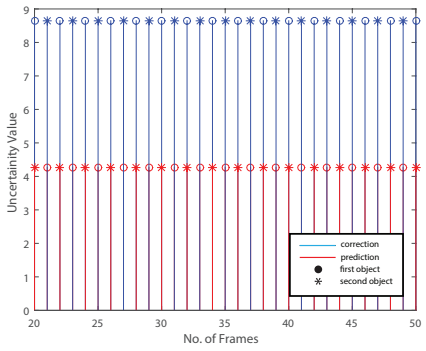
scenario 2



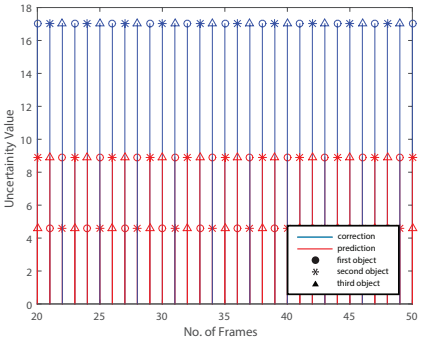
scenario 3



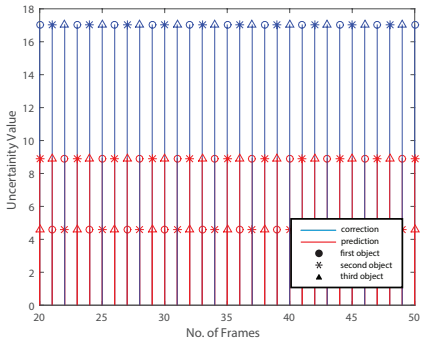
scenario 4



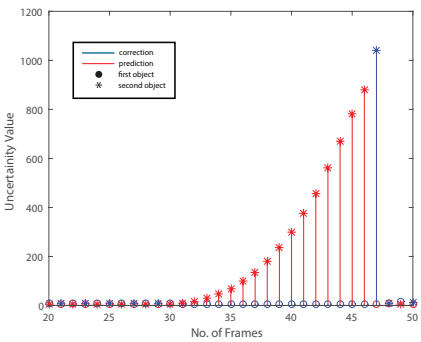
scenario 5



scenario 6



scenario 7



scenario 8

Figure 3.9: Attention Shift using uncertainty information. The prediction is represented by the red colour and the correction is represented by blue colour.

The initial uncertainty is high due to the high initial error covariance matrix. Later on, when the tracker updates the uncertainty information of any target, then the uncertainty becomes low. The uncertainty becomes high again when the tracker alternates its attention on the other targets. This shift of focus between multiple targets is based on the uncertainty computed by the tracker, as discussed in Section 3.2.4 (on page 49).

There is a single target in scenarios 1 and 3, and the uncertainty is low in each frame as the tracker is paying attention to this target only as there is no other object present at that time.

In scenarios with multiple targets, the tracker updates the most uncertain targets first. For example, in frame 20 of scenario 2, the tracker updates the most uncertain target and the uncertainty of the other target increases due to this shift of attention. In frame 21, the tracker again shifts focus to the next most uncertain target.

There is occlusion in frame 31 of scenario 8, as shown in Figure 3.9. The tracker predicts the location of occluded target in frames 31 to 46 and updates another available target. The focus of attention is shifted again to the occluded target in frame 47 when it is detected again with high uncertainty.

The covariance error ellipse shows the region of the set of observations drawn by the state estimation error covariance as discussed in Section 3.2.4 on page 49. The red and blue coloured ellipses represent the prediction and correction step of the tracker, respectively, and the size of these ellipses depends on the tracker's uncertainty as shown in Figure 3.10.

The proposed method is evaluated using the mean square error betweem true and estimated state, as shown in Figure 3.11. The estimated location and true locations are compared by indicating the error rates as shown in Figure 3.11. The target with the lowest error gets the tracker's attentions where other targets having the high error, get no or little attention.

The occluded object in scenario 8, as shown in Figure 3.11, has a high

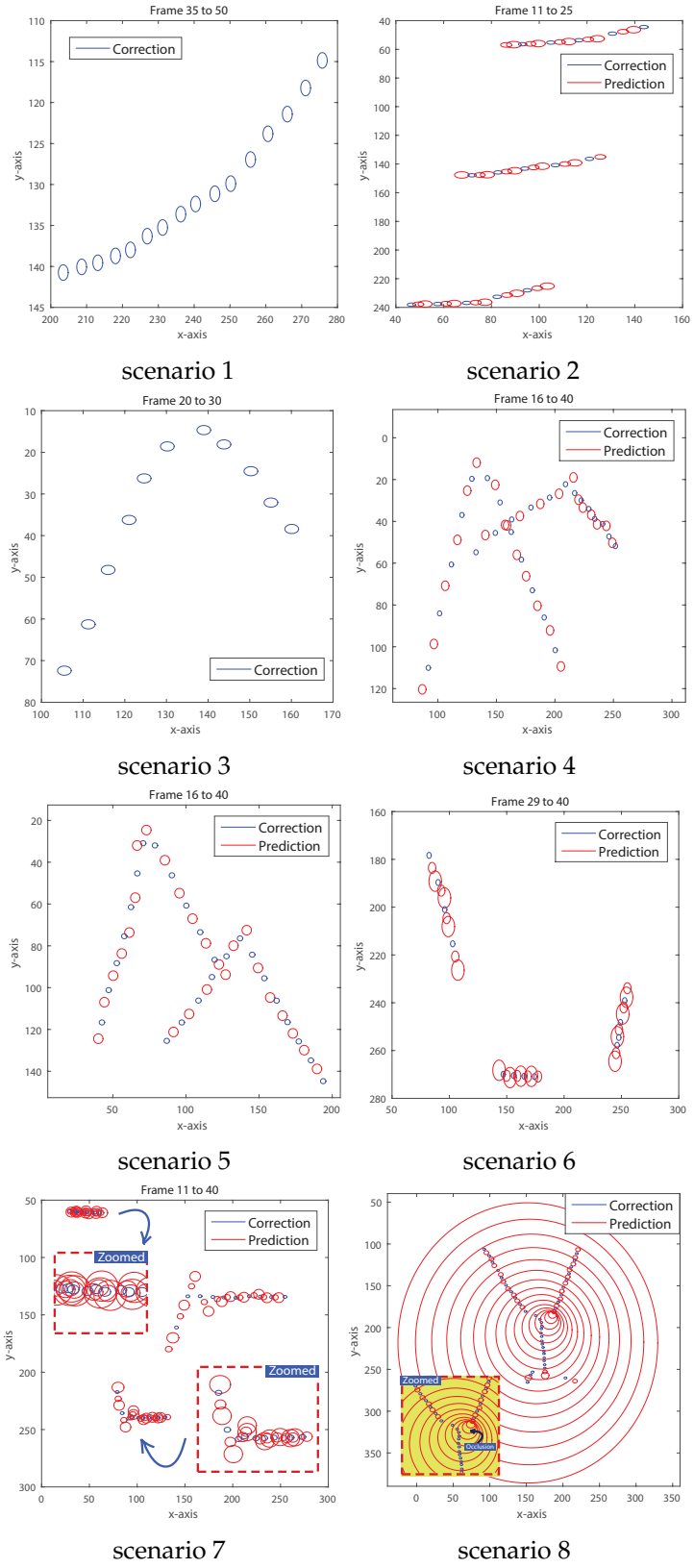


Figure 3.10: State estimation confidence ellipses for the scenarios introduced in Figure 3.6 with the same process noise.

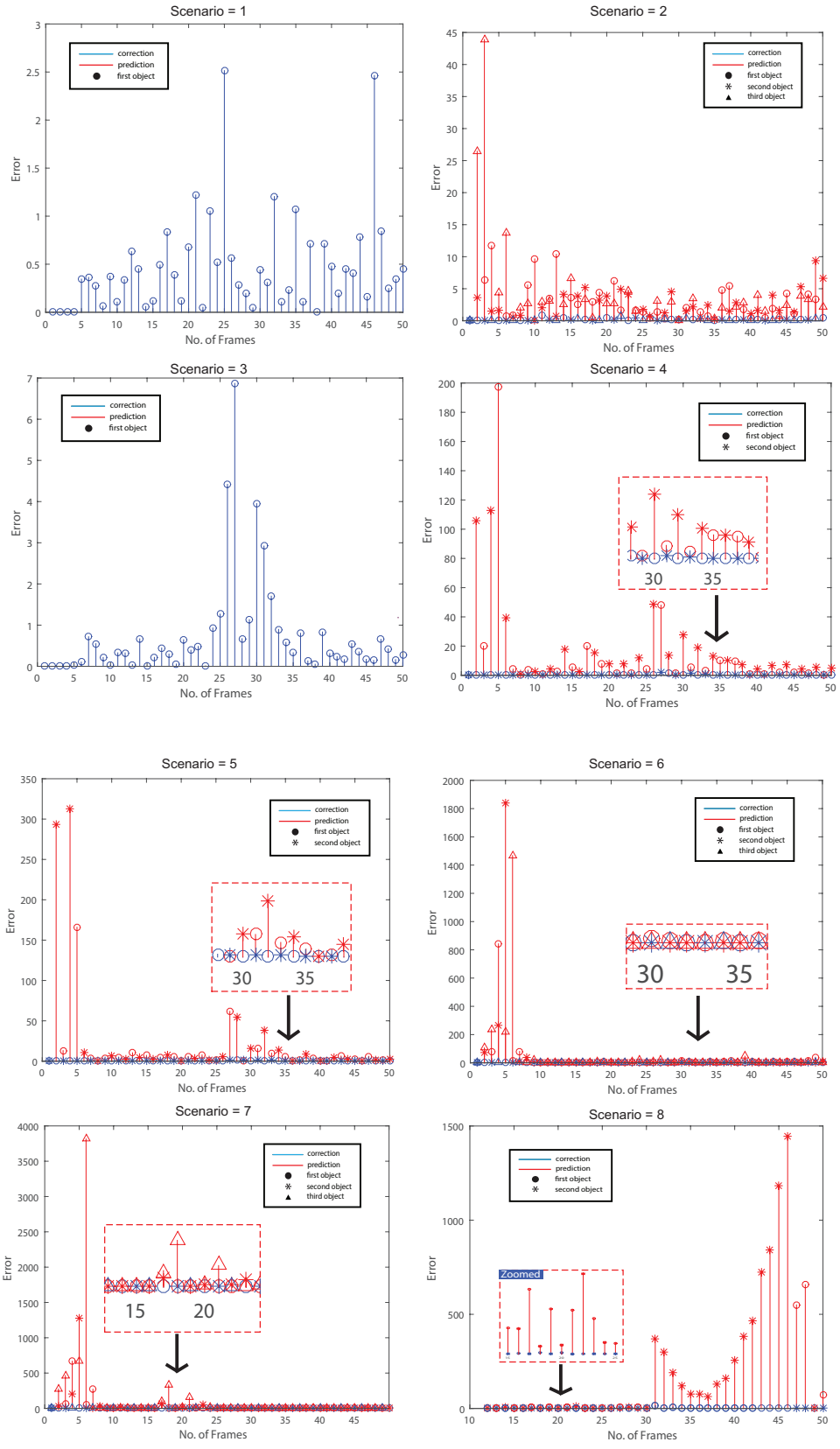


Figure 3.11: Means square error plots for scenarios shown in Figure 3.6.

error as the tracker is predicting the location of this object. The error of this object is minimised when its location is detected and updated by the tracker.

Figure 3.12 shows that the true value is consistent with the estimated location by plotting the response of the tracker. The Kalman gain is shown in Figure 3.13. Initially, due to high error covariance value, the tracker is confident that the measurement is good enough and update the estimate. As a result, the Kalman gain arises initially in each plot. In the later frames, the filter reaches a steady state value.

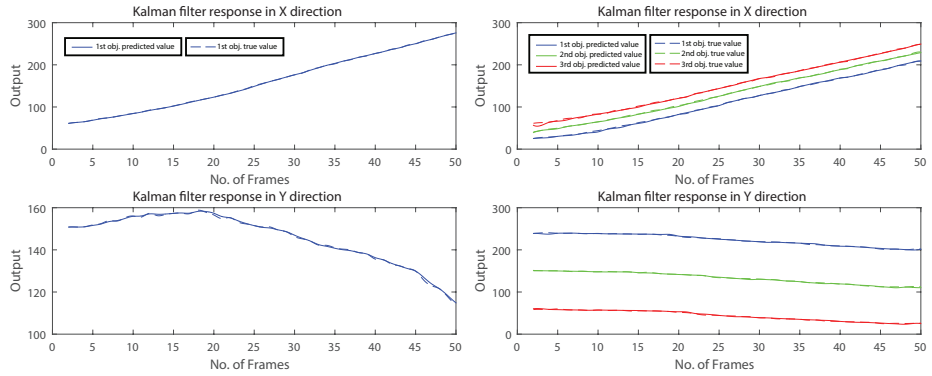
3.6 Process Noise Effects

The process noise plays an important role in determining when the tracker shifts attention from one target to another, as discussed in Section 3.2.4 on page 49. This analysis helps in testing the distribution of attention shift process and can help in the implementation of a real system. Here, a constant velocity motion model is used for the first two simulations and constant acceleration is used for the third simulation as discussed in Section 2.4. The process noise matrices for both motion models are shown in Equations (2.20) and (2.32).

3.6.1 Process noise with the same variances σ_w^2 in x and y direction.

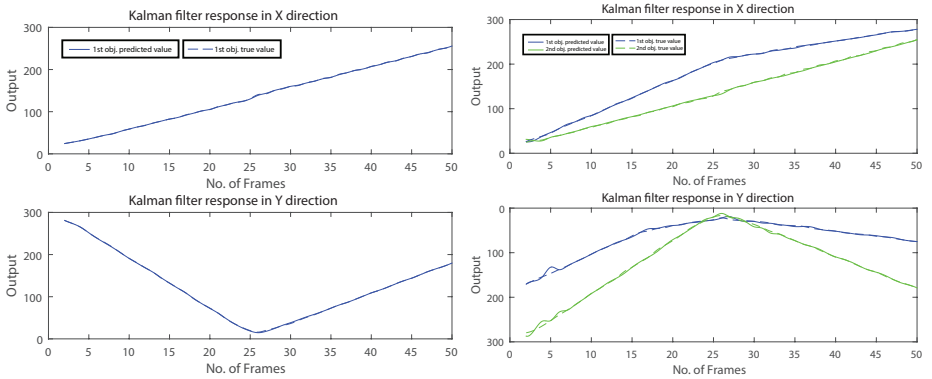
There is a sequential shift in attention, as shown in Figure 3.9 when there is same process noise in the system. If the targets in a scenario have different process noise, then the target with the highest process noise gets the tracker attention as shown in Figure 3.14.

Different process noise is used for each target to analyse the Kalman filter response using scenario 2, as shown in Table 3.3 with the same variances $\boldsymbol{\sigma_w}^2$ in the x and y directions. Here, the process noise affects the



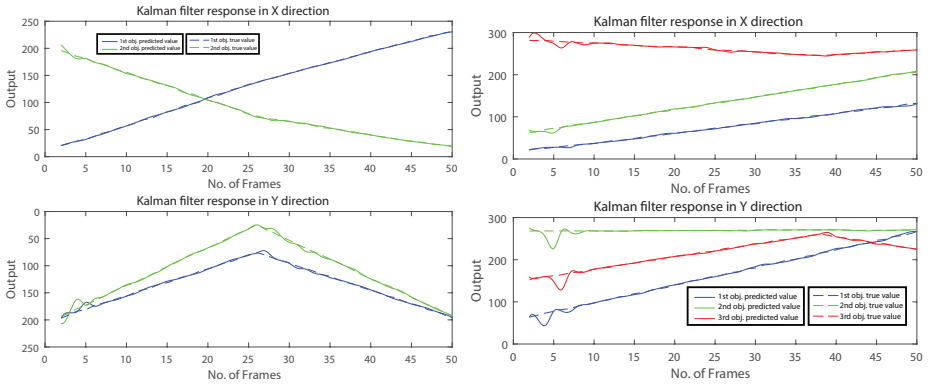
scenario 1

scenario 2



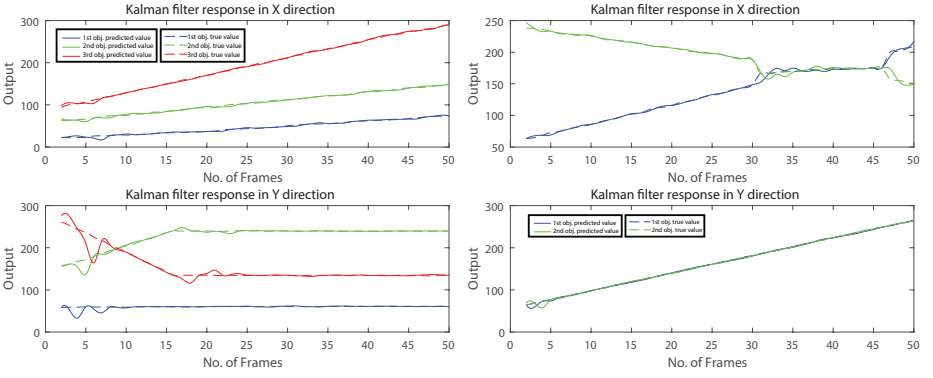
scenario 3

scenario 4



scenario 5

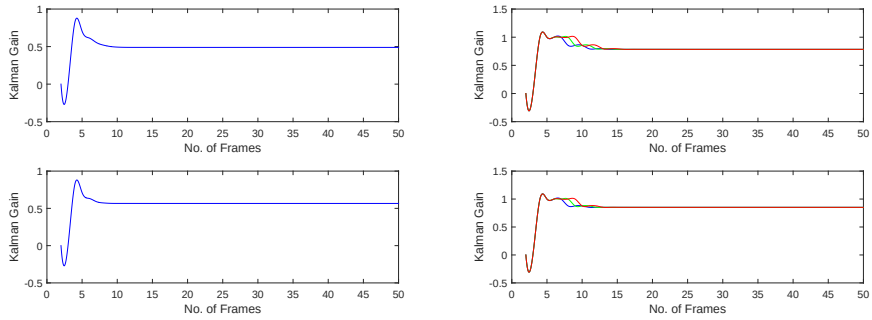
scenario 6



scenario 7

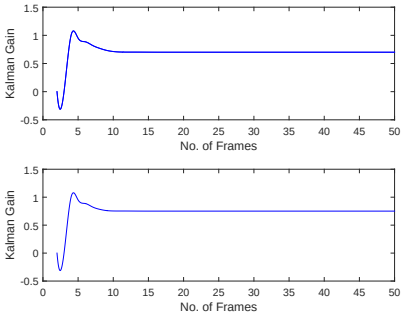
scenario 8

Figure 3.12: Kalman gain output for scenarios shown in Figure 3.6.

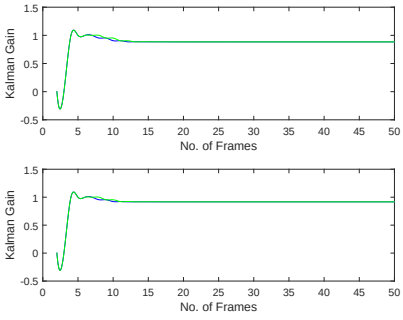


scenario 1

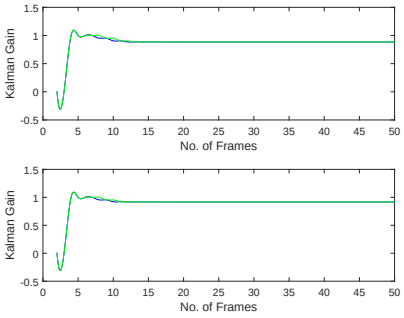
scenario 2



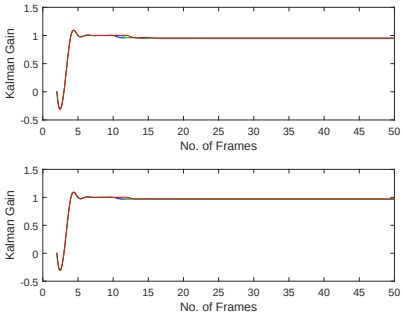
scenario 3



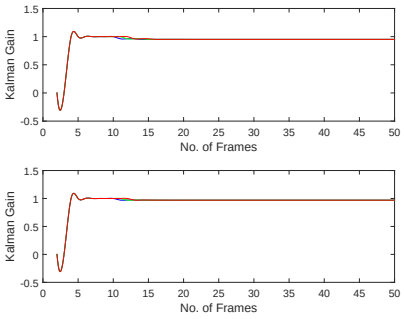
scenario 4



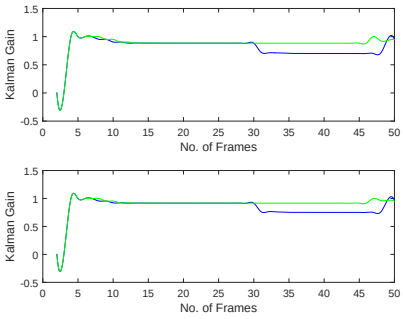
scenario 5



scenario 6



scenario 7



scenario 8

Figure 3.13: Kalman Filter response vs true value of x and y locations are plotted for scenarios shown in Figure 3.6. Two plots drawn for each scenario, i.e., first plot is for x location and other is for y location.

error covariance and uncertainty of the targets. The target with high process noise has more chance of getting the tracker's attention than the other targets.

Table 3.3: Simulation 1: Different variance value is used with process noise

Objects	Variance value	Ellipse shape
1	${\sigma_w}^2 = 10$	Circular
2	${\sigma_w}^2 = 50$	Circular
3	${\sigma_w}^2 = 500$	Circular

3.6.2 Process noise with the different variances ${\sigma_w}^2$ in x and y direction

In the second simulation, for two targets, difference variance value in x and y direction are used for the process noise, as shown in Table 3.4.

As expected, the uncertainty of the target's state estimates having equal variance in both directions grows. The tracker pays most of the attention to this target to minimise its uncertainty. The third target has higher process noise variance in x direction then a y direction. Due to that it gets the second most attention from the tracker as shown in Figure 3.15.

3.6.3 Process noise with the different variances ${\sigma_w}^2$ in x and y direction

In the third simulation, targets follow the constant acceleration motion model, as shown in Figure 3.16 with different variance values as listed in Table 3.5.

The response of the Kalman filter of this simulation is the same as previous simulations. The object with higher process noise gets most of the attention of the tracker.

Table 3.4: Simulation 2: Different variance value is used with process noise in x and y direction.

Object	Variance-x direction	Variance-y direction	Ellipse shape
1	$\sigma_{w_x}^2 = 1$	$\sigma_{w_y}^2 = 25$	Oval
2	$\sigma_{w_x}^2 = 100$	$\sigma_{w_y}^2 = 100$	Circular
3	$\sigma_{w_x}^2 = 500$	$\sigma_{w_y}^2 = 10$	Oval

Table 3.5: Simulation 3: Process noise used for testing Scenario 2 with the different process noise in x and y direction for each object.

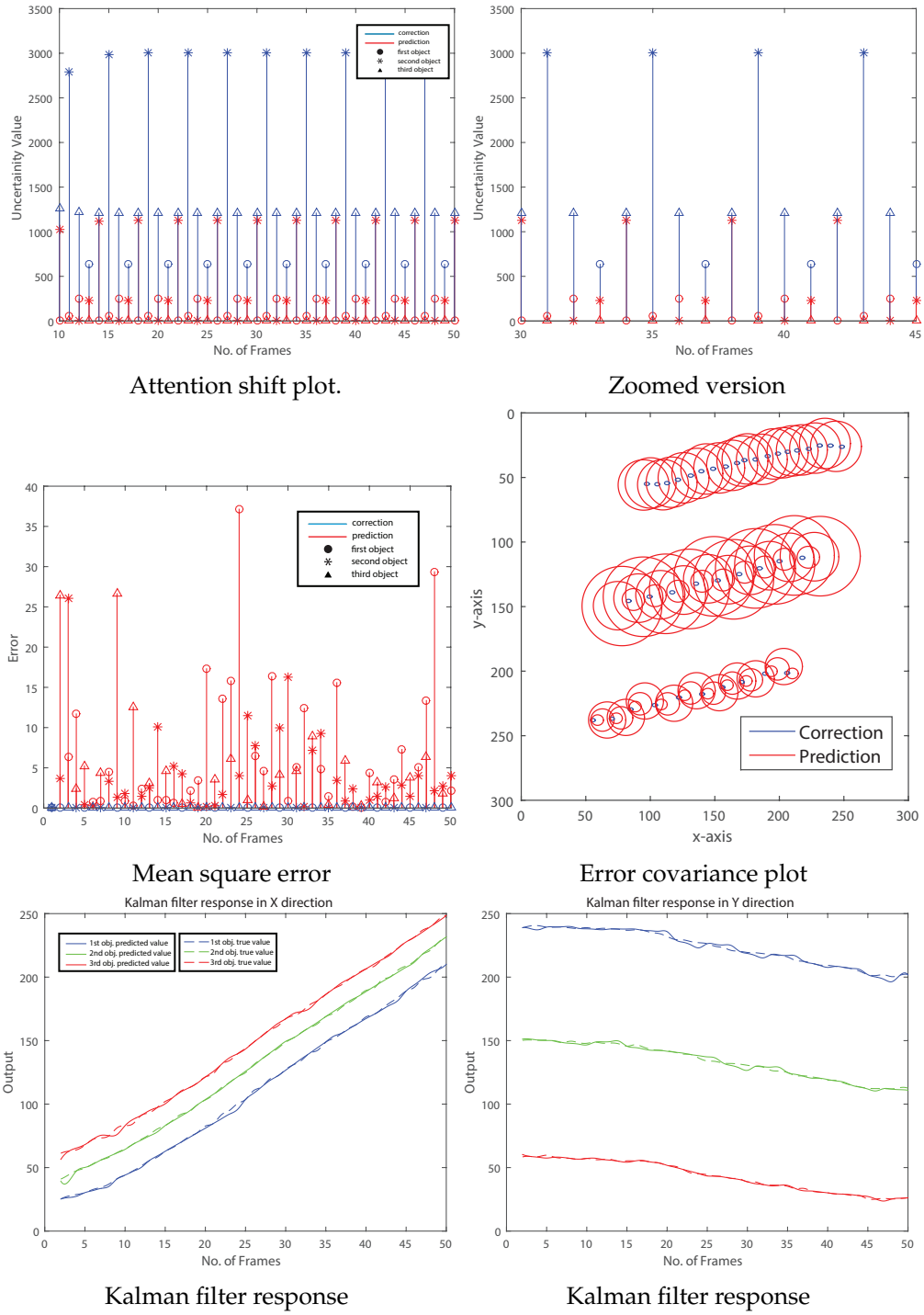
Objects	Process noise	Ellipse shape
1	$\sigma_w{}^2 = 0.2$	Circular
2	$\sigma_w{}^2 = 10$	Circular
3	$\sigma_w{}^2 = 50$	Circular

3.7 Conclusion

In this chapter, a saliency-based multiple object tracking system was developed by combining a visual attention computational model and a Kalman filter. The overall major contribution is to develop an effective multiple object tracking system that can replicate aspects of the human vision such as shifting attention from one target to another in a complex environment.

The proposed method was able to handle occlusion and predict the location of an occluded target. The results show that the mean square error is high between the estimated and the true locations for the occluded object as the tracker is not paying attention.

Experiments with different scenarios, as discussed in this chapter show that the proposed tracking system effectively shift attention from one target to another. Currently, we have used the RGB-based saliency method to detect objects. In the next chapter, we compared two colour spaces that can be used with saliency method.

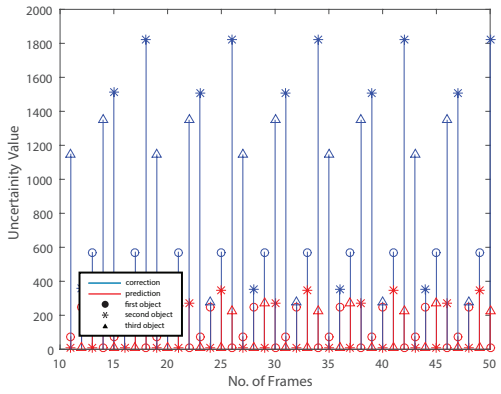


Attention shift plot. Zoomed version

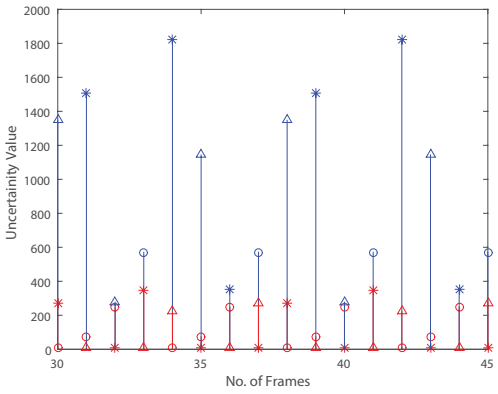
Mean square error Error covariance plot

Kalman filter response Kalman filter response

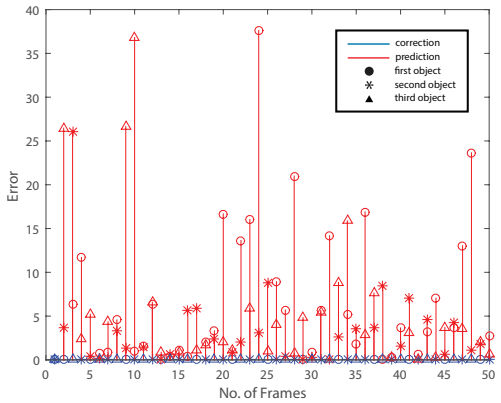
Figure 3.14: Kalman filter output for scenario 2 having three objects with different process noise as shown in Figure 3.6.



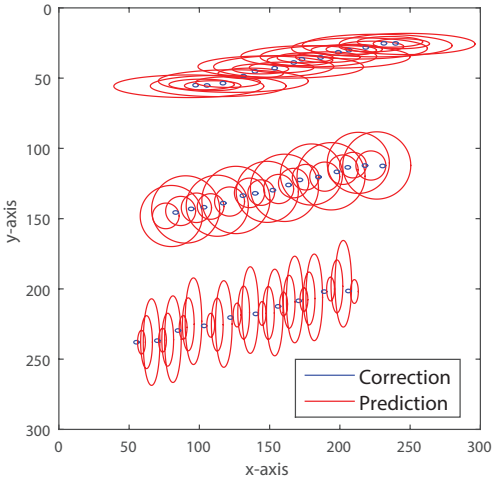
(a). Attention shifting using uncertainty.



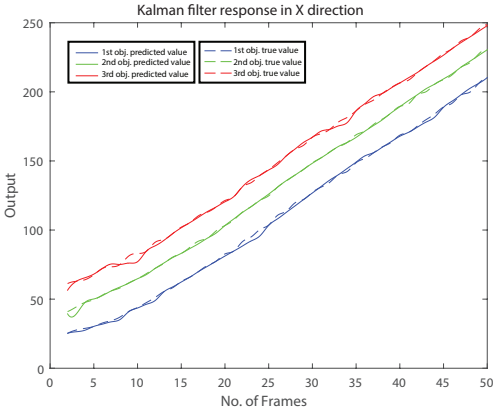
(b). Zoomed version of attention shift plot.



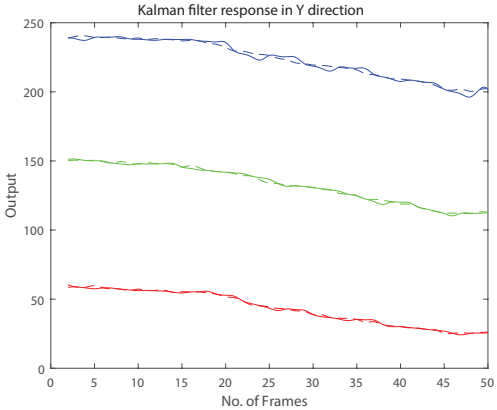
(c). Mean square error.



(d). Error covariance plot.

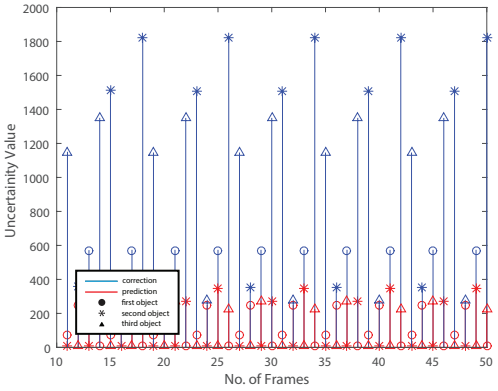


(e). Kalman filter response in x direction.

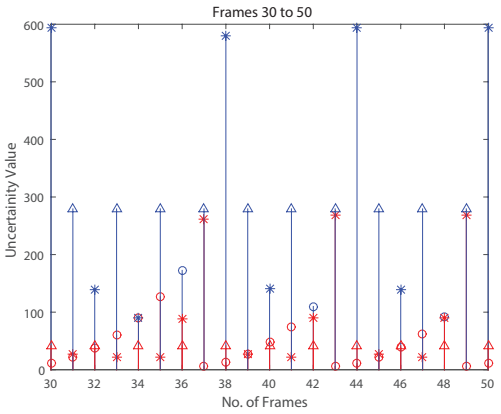


(f). Kalman filter response in y direction.

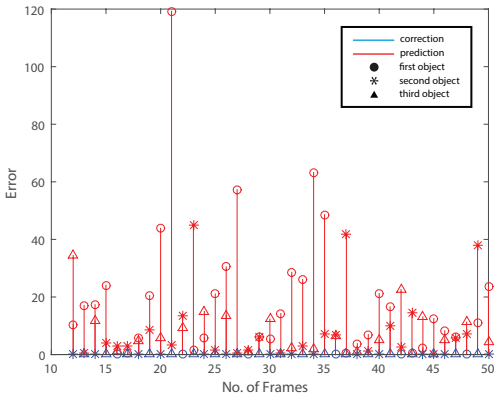
Figure 3.15: Kalman filter output for scenario 2 having three objects with different variance value in x and y direction as shown in Figure 3.6.



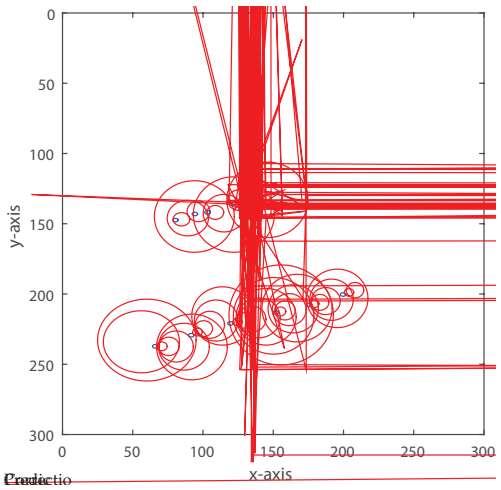
((a). Attention shifting using uncertainty.



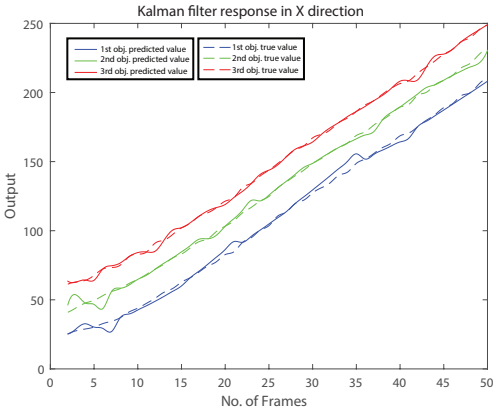
(b). Zoomed version of attention shift plot.



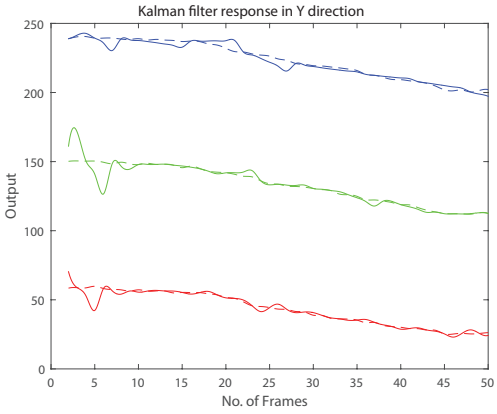
(c). Mean square errorError covariance plot.



(d). Error covariance plot.



(e). Kalman filter response in x direction.



(f). Kalman filter response in y direction.

Figure 3.16: Kalman filter output for scenario 2 having three objects with different process noise in x and y direction This scenario is shown in Figure 3.6.

Chapter 4

Motion Saliency

The human vision system (HVS) pays more attention to the moving objects than the static areas. Due to this fact, motion becomes one of the important features of the visual attention model. The overall objective of this chapter is to integrate motion feature to the visual computational model and incorporated into an object tracking system to detect salient moving objects from video. To implement this objective, several sub-features such as colour, colour change between different frames, optical flow estimation and background subtraction are explored, and the appropriate feature combinations are chosen for further experimentation.

4.1 The Proposed Model: Motion-Based Saliency Detection (MSD)

This section describes how a visual attention model performs efficient detection of moving objects. The proposed motion-based saliency detection (MSD) model is shown in Figure 4.1 that contribute towards building the final object tracking system. As the figure shows, the first step involves the conversion from the RGB colour space to the HSV colour space. This conversion can help specify the colours as HSV colour space is non-linear and

represent the human perception. In the second step, several mid-level features are extracted to generate the saliency map (SM) using the pyramidal approach of Itti's feature model [14].

The reason for using this low-level feature-based model is because it is computationally more efficient than other models that use high-level features for saliency detection. The low-level features of Itti's original model include colour, intensity and orientation but no feature that relates to motion. For motion analysis, we have added new mid-level features to the basic colour feature of this model. The reason for adding new features is to generate a quality saliency map and to demonstrate our proposed motion saliency mechanism. By adding these new features, a framework is developed for motion-based saliency detection model that can be used for target feature detection.

The proposed MSD model is divided into three main tasks:

1. **Colour Space Conversion**
   The red (R), green (G), blue (B) channels of each frame are converted into the equivalent Hue (H), Saturation (S) and Value (V) components. Detailed analysis on these colour spaces is done in Section 4.2.

2. **Feature Extraction**
   In this second step, the new motion detection based mid-level features is computed along with a modified low-level colour feature based on Itti's feature model. The detailed implementation and performance of these features are explained in Section 4.3, but in brief:

   (a) The first feature (green box in Figure 4.1) is the modified version of the basic colour feature of the Itti's attention model. We threshold the components of HSV frames to find the specific coloured targets. Thresholding is done by selecting the required coloured values from the Hue channel.

   (b) The second feature (grey box in Figure 4.1) includes several sub-features for motion analysis, namely colour change, optical flow

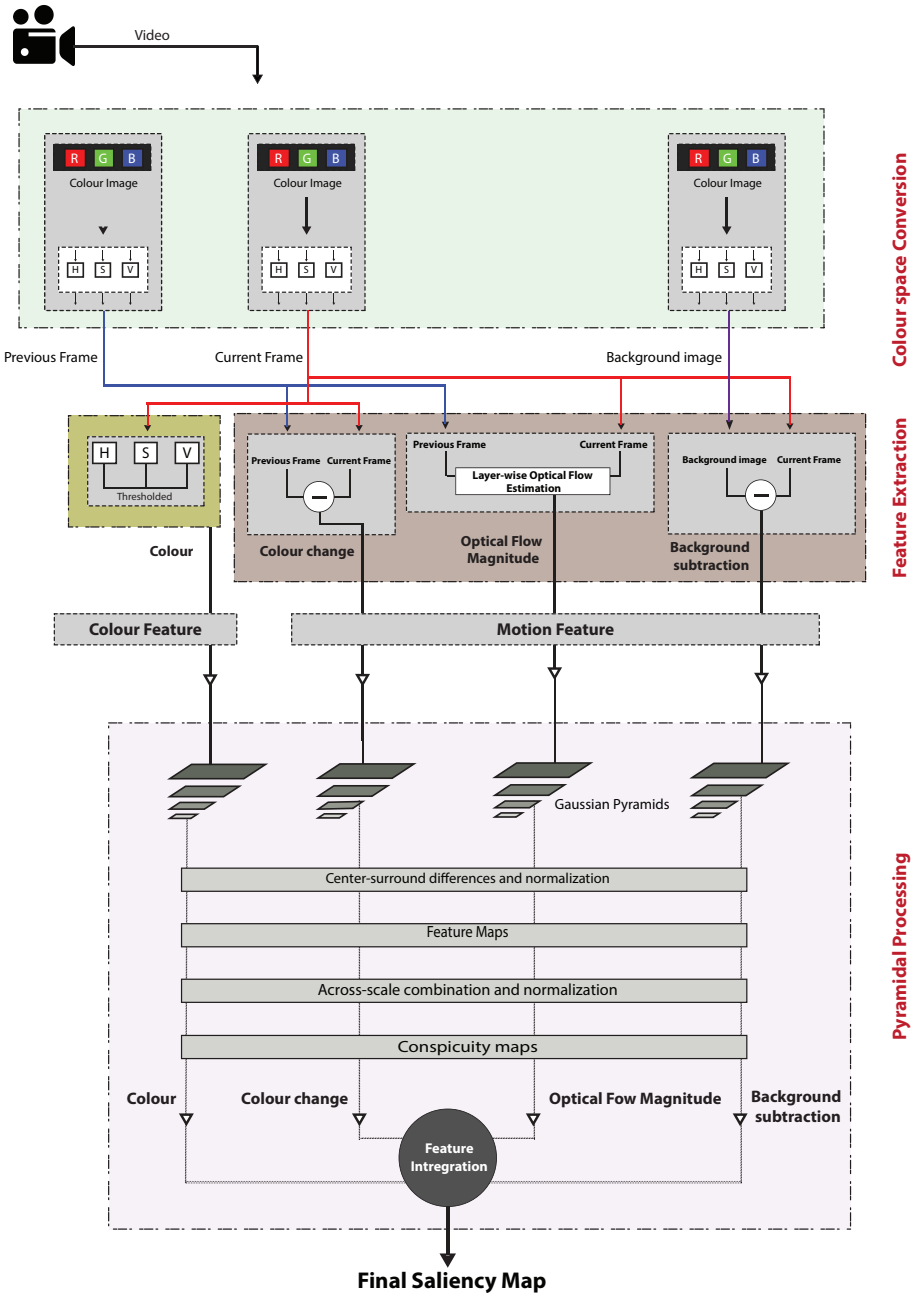


Figure 4.1: The proposed MSD model.

and simple background subtraction.

i. The first motion sub-feature is the colour change feature based on the temporal differencing (i.e. the current frame is subtracted from the previous frame to detect the HSV-based colour difference between the consecutive frames). The result of this feature extraction are the salient moving regions from the frames.

ii. Optical flow is the representation of the displacement of pixels in video sequences. The current and previous HSV frames of the scenario are used to estimate the flow using the layer-wise method by creating a mask. The detailed analysis on the optical flow feature is given in the Section 4.4.2.

iii. A simple background subtraction technique is incorporated as the third feature to analyse motion. It has been computed using a reference image either selecting manually or averaging several previous frames. This reference image is subtracted from the current image to detect moving object.

3. **Pyramidal processing**
   The third step is the computation of the motion saliency map. We have used pyramidal approach of Itti et al. [14] model that uses a Gaussian filter applied after the computation of the features to produce the feature map using a center-surround mechanism. Finally, a saliency map which represents the saliency is generated by the combination of the conspicuity maps.

To achieve the main steps of the proposed system, we have performed the following sub-tasks:

1. We have created several datasets, each with multiple objects in controlled scenarios with variability in the object's location. To make these scenes close to real-world situations, we have included complicating factors such as changes in illumination, object occlusion

and moving camera. We have recorded manually drawn the ground truth of these datasets to evaluate the performance of detection.

2. We conducted some preliminary experiments to compare the HSV and RGB colour spaces when studying the main colour and motion feature to identify which one is the best to use with the proposed detection model. As a result, the colour space of the visual attention computation model in [14] is modified and converted from RGB to HSV colour space. The detailed analysis of this modification is giving in Section 4.2.

3. Different features such as colour, colour change, optical flow and background subtraction are used to detect motion as discussed in this chapter. The colour feature is thresholded to a specific colour to detect known coloured targets only. The colour change feature uses the temporal differencing method that uses consecutive frames to identify motion. Optical flow is another popular method to estimate flow between consecutive frames caused by the relative motion of the observer and the scene. Here, we used flow information to detect motion saliency in challenging situations such as camera moving and illumination changes.

   Background subtraction is used as one of the features because it can adapt changes in background, i.e., illumination changes in the subsequent frames.

4. To demonstrate the computation of the saliency map we have compared two approaches. The first is the Gaussian pyramidal approach as in the work of Itti et al. [14] while the second is a non-pyramidal processing approach. The pyramidal approach is sensitive to the presence a local region that is different from its surrounds, while the non-pyramidal approach is sensitive to the presence of a feature itself. In this chapter, each feature is computed using the both approaches and the results compared with each other to evaluate

performance. To test the performance of the system without pyramidal processing, we created the saliency map directly from the colour and motion features (essentially, removing the entire low box in Figure 4.1 except for the feature integration module.). The non-pyramidal approach saves the extra computation required for finding the salient feature without any fixation mechanism.

5. We integrated the features to analyse the performance of the system. The integration of the saliency maps from the different features is performed using arithmetic operations such as addition and multiplication.

6. The area under the Precision-Recall curve [96] metric is used to evaluate the quality of a motion saliency maps using different threshold values. Precision is defined as $TP/(TP + FP)$, where TP is true positive and FP is false positives. Recall is defined as as $TP/(TP + FN)$, where FN false negatives. If the curve is higher and having low Recall values, then the saliency map is considered as good because it identifies more salient locations with respect to ground truth. In our work, the curves of various features are compared with a ground truth map to evaluate performance.

4.2 Colour Space Analysis

One of the important features for a visual attention model to compute saliency is colour. Colour space is a specific representation of colour and has an important role in defining the overall target's appearance and detect salient regions for image understanding and analysis [97]. Different methods have been developed that used the colour feature to detect a salient object from a scenario [97, 41]. For the proposed motion saliency system, we would like to use the colour space that can generate saliency maps effectively, regardless of challenging situations. Here, we have given

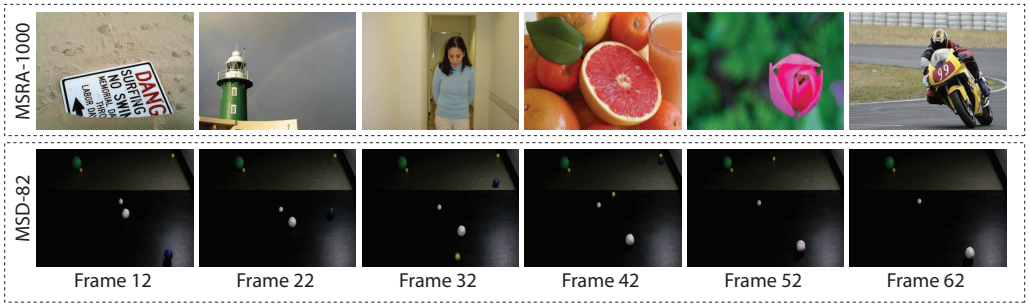


Figure 4.2: Sample images (a) MSRA-1000 (b) MSD-82.

a detailed analysis by defining the overall image content in term of the RGB or the HSV colour spaces to be used with the proposed system.

Experiments were carried out on different datasets to measure the performance of object detection using HSV colour space instead of RGB colour space. The effectiveness of these experiments is evaluated using the Precision-Recall curve, which is used to compare the quality of a saliency map with ground truth map. A good saliency map is the one that exhibits the high similarity with the ground truth.

4.2.1 Experimental Evaluation

To evaluate and generate the saliency maps, we have used two datasets. Examples from each dataset are shown in Figure 4.2 that are explained as follows:

1. **MSRA-1000 Dataset**
   The first dataset is the MSRA-1000 that we have used to calculate the Precision-Recall curve over the entire 1000 image saliency dataset. This dataset is a subset of the MSRA-5000 dataset with the accurate human-labelled ground truth of saliency regions [98]. This dataset has various types of contents with different simple backgrounds.

2. **MSD-82 Dataset**
   We have created this dataset having 82 frames in which different

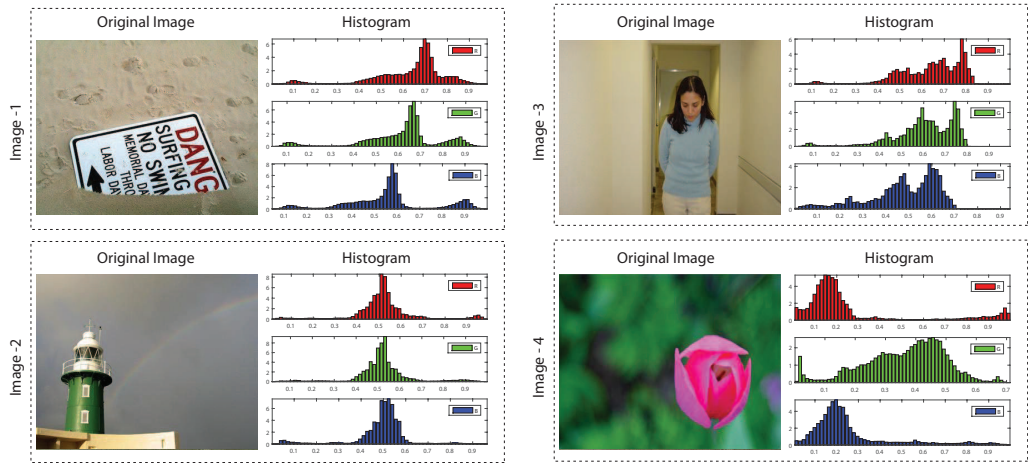


Figure 4.3: Sample images with the histogram represeting the individual channels of RGB colour space.

moving or stationary objects of various sizes are present. The main purpose of creating this scenario to have a very controlled dataset with variability in the objects. This scenario is recorded indoor with a simple background. The ground truth is created with all the objects are marked as salient in this dataset.

4.2.2 RGB colour space analysis

RGB model is a non-uniform colour space in which each colour is described by spectral components of red, green and blue. These components are perceptually non-uniform as the Euclidian distance between colours in the RGB space does not correspond to colour differences perceived by humans [70]. High correlation is present among these components because of the dependence on intensity [71].

Itti et al. [14] used the RGB colour space to extract the Red, Green, Blue and Yellow colour channels. Further, these extracted colour channels were normalised to decouple hue from the intensity. The Red-Green and Blue-Yellow colour opponencies were created for the colour channels. These

opponencies are created to replicate the human vision as HVS processes colour information by assuming the Red-Green and Blue-Yellow colour opponencies. The following equations form the different channels of RGB colour space using red (r), green (g) and blue (b) components:

$$\boldsymbol{R} = \boldsymbol{r} - \frac{(\boldsymbol{g} + \boldsymbol{b})}{2} \tag{4.1}$$

$$\boldsymbol{G} = \boldsymbol{g} - \frac{(\boldsymbol{r} + \boldsymbol{b})}{2} \tag{4.2}$$

$$\boldsymbol{B} = \boldsymbol{b} - \frac{(\boldsymbol{r} + \boldsymbol{g})}{2} \tag{4.3}$$

$$\boldsymbol{Y} = \frac{(\boldsymbol{r} + \boldsymbol{g})}{2} - \frac{(\boldsymbol{r} - \boldsymbol{g})}{2} - \boldsymbol{b} \tag{4.4}$$

$$\boldsymbol{RG} = (\boldsymbol{R} - \boldsymbol{G}) \tag{4.5}$$

$$\boldsymbol{BY} = (\boldsymbol{B} - \boldsymbol{Y}) \tag{4.6}$$

Here, the r, g and b are the raw pixel pf red, green and blue channel's data whereas R, G and B are the calculated normalised colours. The opponent colours Red/Green and Yellow/Blue are simply defined as the difference between the red, green, blue or yellow colours. In the visual cortex, the set of opponent colours represents a "colour double-opponent" system. One colour (e.g., red) excites the neuron in the centre and the other colour (e.g., green) inhibits, whereas the converse is true in the surround [14].

The components of the RGB colour space are highly correlated because the change in these components depends on the intensity change. If the intensity changes in an image then all three components will change accordingly [71].

First, we have generated the normalised histograms of MSRA-1000 dataset as shown in Figure 4.3 to check the tonal distribution of the individual channels by plotting the number of pixels for each tonal value.

Each histogram describes the tonal distribution of each channel in an image by plotting number of pixels for each tonal value. The horizontal axis of the histogram represents the distribution by characterising the left side as the dark regions, middle represent the grey, and right-hand side of

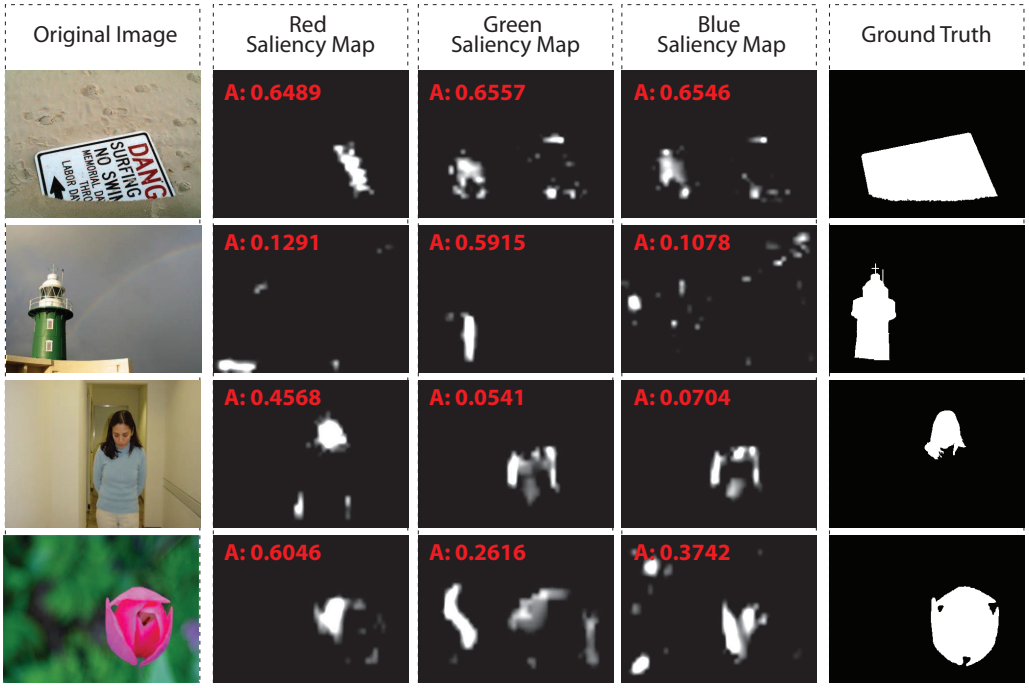


Figure 4.4: Sample images from the saliency maps using MSRA-1000 dataset.

the axis represents the white areas in an image. The histogram of the sample images shows that the components of RGB colour space have similar distributions.

We have then created saliency maps using the RGB colour space to see the effect of the individual channels in defining the overall salience of the image content. We have used the red coloured value on the saliency maps to generate the Precision-Recall score. The MSRA-1000 dataset, as shown in Figure 4.4 has shown a good Precision-Recall score mentioned as for the Red saliency map as compared to the other channels. This is because this dataset has many red coloured salient objects. Whereas the MSD-82 dataset as shown in Figure 4.5(a), has shown better Precision-Recall score for the Green component than the Red component. The Red component performs poorly in defining the saliency for this dataset as some patches of

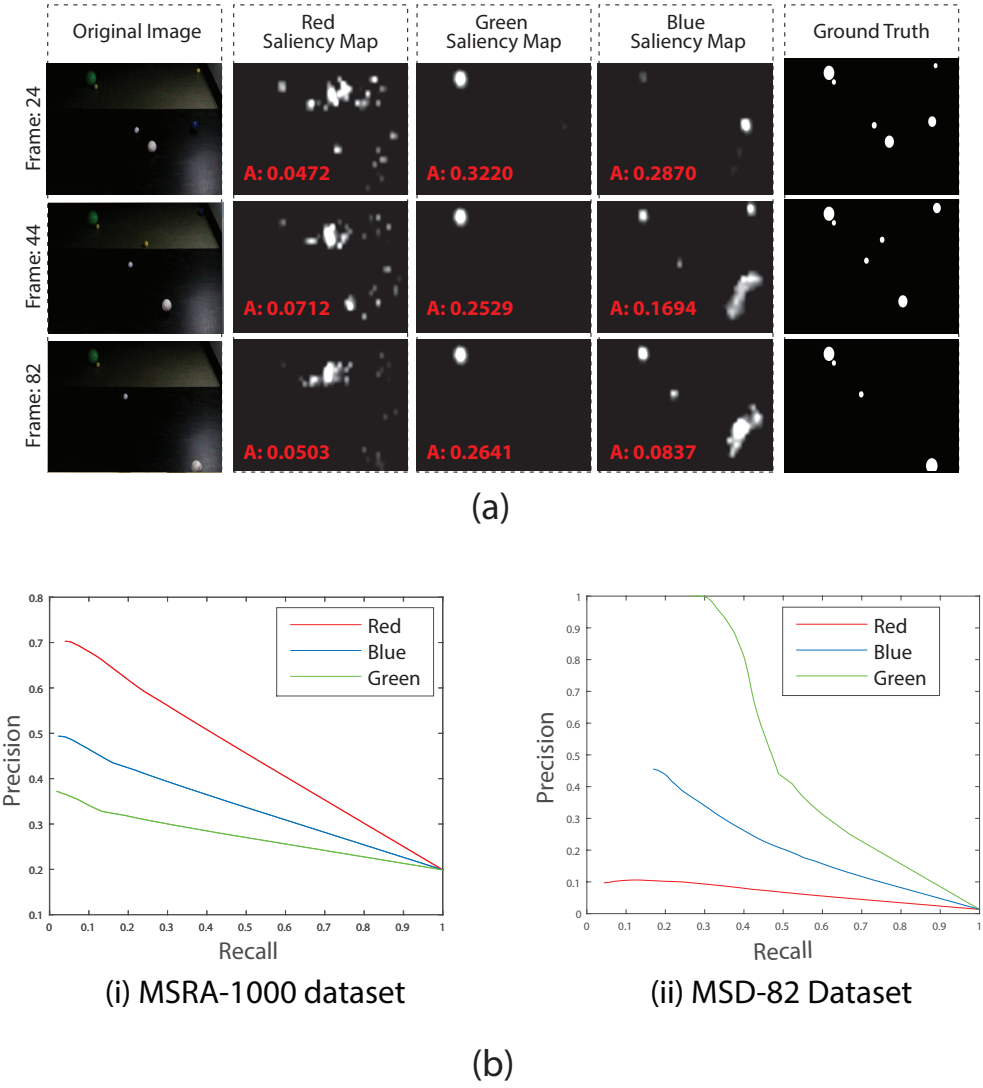


Figure 4.5: Sample images with saliency maps (a) MSD-82. (b) Precision-Recall performance evaluation.

the background are closer to red. With this analysis, we can notice that the individual channels of RGB colour space perform according to the colour information present in an image. This can be seen using the Precision-Recall curve for the RGB individual channels is shown in Figure 4.5(b) where Red channels perform higher for the MSRA-1000 dataset as com-

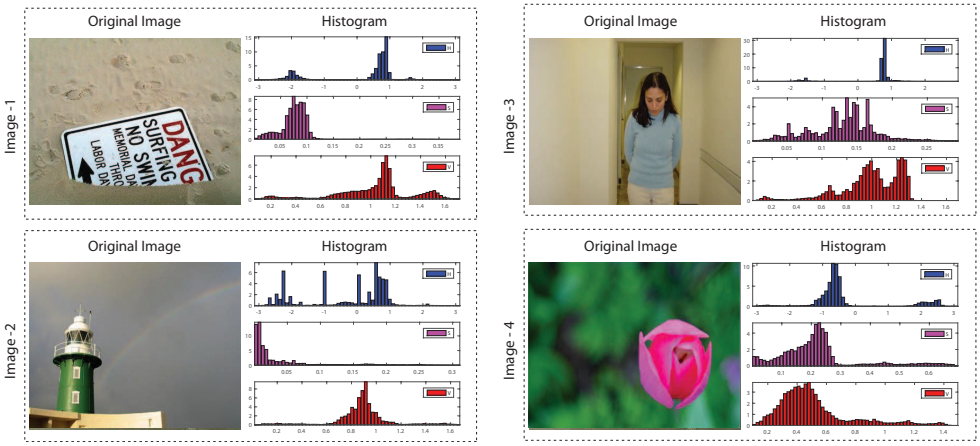


Figure 4.6: Sample images with the histogram representing the individual channels of HSV colour space.

pared to another dataset where red colour content is not salient.

4.2.3 HSV colour space analysis

A non-linear HSV colour space represents the human perception using Hue (colour), Saturation (colour depth) and Value (brightness) components that can specify colour in an intuitive manner. The desired hue can be selected and modified using the specific values of the saturation and value. It separates the luminance component from the colour information that is helpful in the various object detection applications [69, 5].

In the proposed system, we have used HSV instead of RGB colour space. Hue represents the colour; Saturation describes the colour depth and depends on the amount of white light mixed with Hue. Value is the brightness of the colour. Saturation and Value described in percentages from 0 to 100%. The equations that are used to convert the RGB colour

space into the HSV colour space are as follows:

$$\boldsymbol{V} = \max(\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}) \tag{4.7}$$

$$\boldsymbol{S} = V - \min(\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B})/V \tag{4.8}$$

$$\boldsymbol{H} = \frac{\boldsymbol{G} - \boldsymbol{B}}{6S}, \qquad if \quad \boldsymbol{V} = \boldsymbol{R} \tag{4.9}$$

$$\boldsymbol{H} = \frac{1}{3} + \frac{\boldsymbol{B} - \boldsymbol{R}}{6S}, \qquad if \quad \boldsymbol{V} = \boldsymbol{G} \tag{4.10}$$

$$\boldsymbol{H} = \frac{2}{3} + \frac{\boldsymbol{R} - \boldsymbol{G}}{S}, \qquad if \quad \boldsymbol{V} = \boldsymbol{B} \tag{4.11}$$

First, we have generated normalised histograms to check the tonal distributions of the individual HSV component. As shown in the Figure 4.6, the information of the colour, shade and brightness are not correlated and can be separated. The histograms of the Hue and Saturation show the even distribution and spread over the horizontal axis.

The Value component's histogram shows the significant rise in the middle of the axis as it represents brightness in the image. These channels can be used as the individual component or in combination for detecting the salient regions from an image. We can detect specific targets by using the specific colour values of these channel's histograms. The experiments on thresholding for detecting specific target is explained later in this chapter for the target feature detection.

Then, we have generated the saliency maps using HSV colour space using Itti model [14] to analyse the performance of individual component of the HSV colour space in identifying the saliency from an image. As shown in Figure 4.7, the Value component of the HSV colour space has given higher precision recall score by detecting the brighter region more effectively than the Saturation or Hue components.

The Itti model [14] using the bright salient areas information to shift the focus of attention after a series of normalisations that promotes feature maps with a small number of the strong peak while suppresses other maps in an image. The Hue component performs poorly as it detect irrelevant colour information from the background.

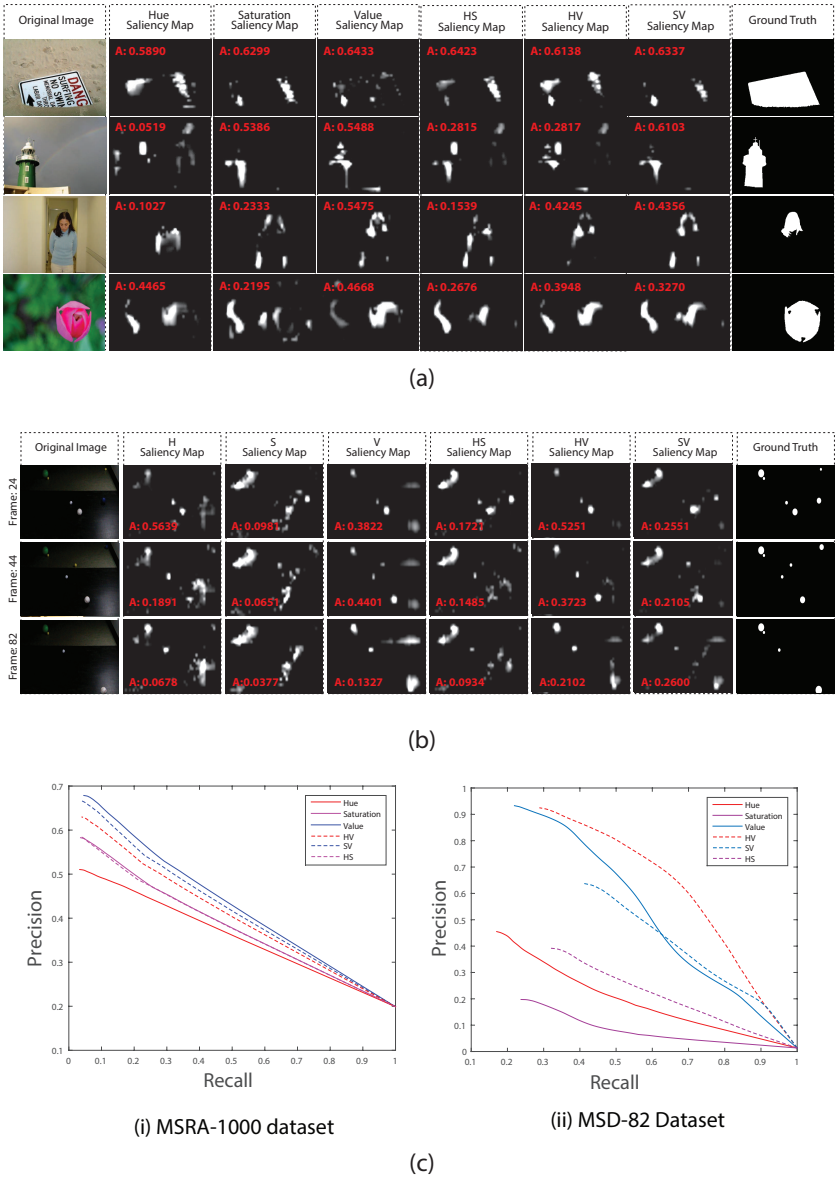


Figure 4.7: Sample images with saliency maps (a) MSRA-1000 dataset with Histograms for each channel (b) MSD-82 Dataset. (c) Precision-Recall performance evaluation

The saliency maps generated using the combination of different channels of HSV colour space such as Hue-Saturation, Hue-Value and

Saturation-Value components have shown better performance compared to the use of the individual components. As shown in Figure 4.7, the Saturation-Value combination shows high Precision-Recall score in detecting the brighter shaded salient objects whereas the Hue-Value combination also achieved high Precision-Recall score while identifying the bright salient region using the Hue information as shown in Figure 4.8. In both cases, the Value component played a significant role while detecting the intensity using the Hue or Saturation information.

We have tested the HSV-based Itti model on MSD-82 dataset as shown in Figure 4.7. This analysis is necessary to check if we get a similiar performance by using the HSV colour space for generating the saliency maps as with the MSRA-1000 dataset. This current dataset has different coloured objects. The Value component gives high performance as shown in Precision-Recall curve as it detects the brighter region from the frames better than the other components of the colour space. When the Value component is combined with Hue and Saturation similar good performance can be seen from the same plot.

The Hue and Saturation component, give poor performance when analysed individually as these elements detect irrelevant colour information from the background. From these experiments, we can suggest the Value component as an essential part in detecting saliency areas in from the images.

If we combined all channels HSV colour space and RGB opponencies, we can notice using Figure 4.8 that if HSV colour space is used with all its channels then it gives a higher Precision-Recall score in detecting saliency then the highly correlated RGB colour space. This analysis confirms our intuition that HSV colour space should be more robust to lighting changes than the RGB colour space. Especially, the Value component and the combination of the Value components with another HSV components can improve the detection of the salient objects from images, when compared with the RGB alternative.

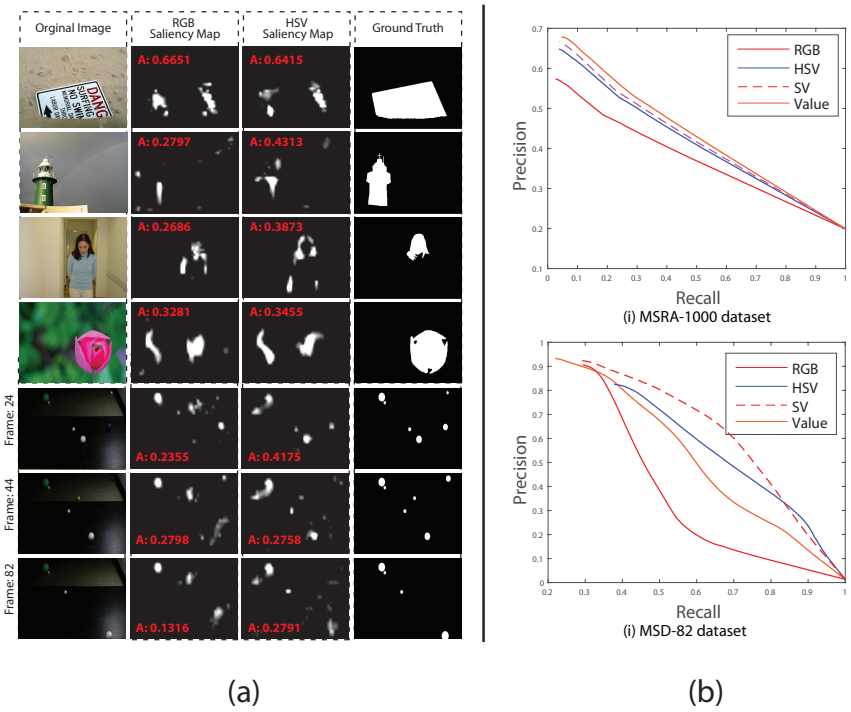


Figure 4.8: Sample images from the result of analysis of combination of RGB and HSV using the MSRA-1000 and MSD-82 datasets.

4.2.4 Non-Pyramidal Processing

In the previous experiments, we have used the pyramidal approach of Itti et al. [14] to extract the most relevant irregular visual features using the local contrast by finding then the difference between a region and its surroundings. Here, Itti used a low-pass filter and sub-sampled the input image by a factor of 2 at each stage, forming a dyadic Gaussian pyramid for each feature in this model.

The main advantage of using Gaussian pyramids is the efficient approximation of center-surround contrasts. This center-surround local contrast extracts the salient objects at various scales. Normalisation is used in the pyramidal approach to suppress the different centre-surround con-

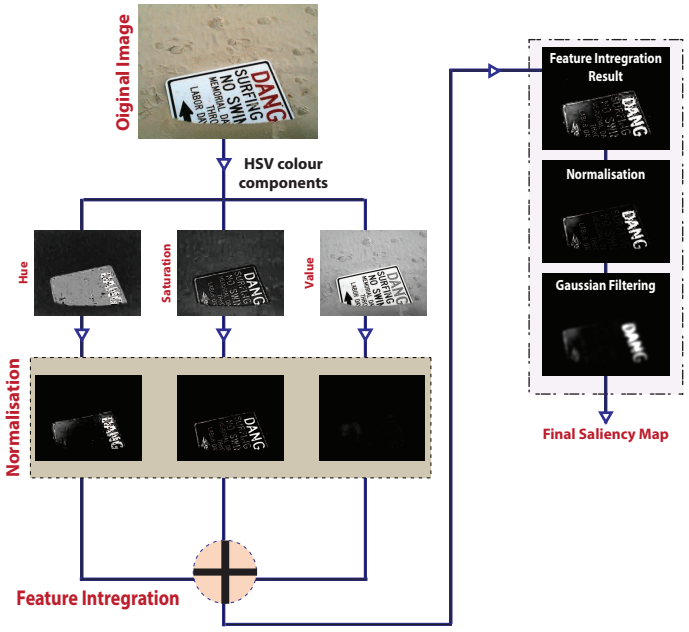


Figure 4.9: Saliency Map generation using the non-pyramidal approach.

trast maps with different peak responses. In our experiments this technique extracts unusal colours while suppressing feature maps with many comparable peak responses and promotes feature maps with few strong peaks.

In this section, we are interested in testing the performance of the feature extraction with the top-down non-pyramidal approach to extract any salient colour. The HSV components are extracted using the original image as shown in the model diagram in Figure 4.9. These components are normalised using iterative normalisation algorithm as proposed by Itti et al. [14] to values between 0 and 1 and then iteratively convolved by a large 2D difference of Gaussians (DoG) filter. After each iteration, negative results are discarded. The DoG based normalisation is highly effective in supressing unwanted noise and strengthening the strong peaks.

The three normalised HSV colour space features are summed to produce a final saliency map. Here, we have used simple summation for

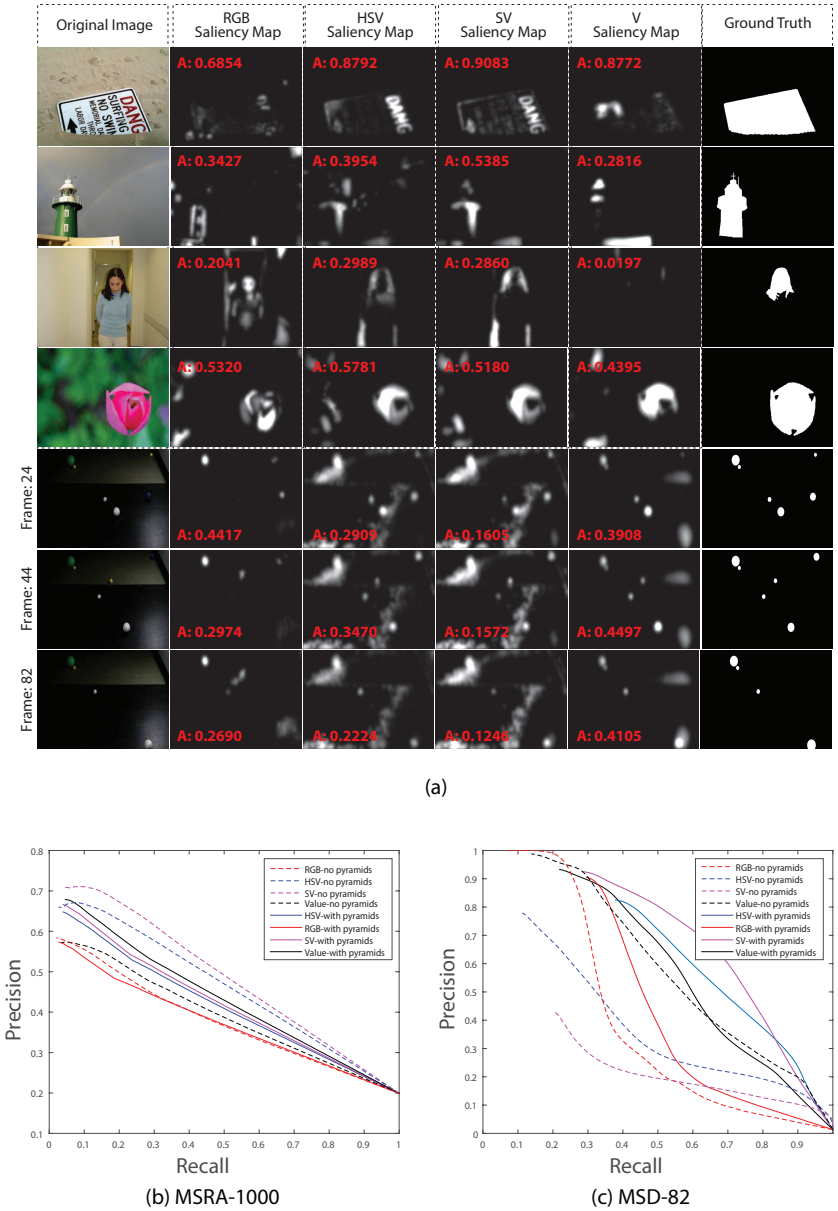


Figure 4.10: Sample images (MSRA-1000 and MSD-82 datasets) by using non-pyramidal approach (a). Precision-Recall performance evaluation (b) and (c).

integration as the three HSV components are highly uncorrelated. This integration result is further normalised using the same Gaussians (DoG) filter to suppress any further unwanted noise.

After the final normalisation, a Gaussian filter is used to smooth the result to reduce noise and enhance the overall image. We have used an appropriately sized Gaussian filter with a standard deviation of 8 pixels and kernel size 65×65 pixels by selecting manually.

As shown in Figure 4.10, the results are similar to our previous pyramidal approach. The saliency map generated by the RGB non-pyramidal approach shows lower Precision-Recall curve than the HSV colour space due to the high correlation between the RGB components as previously discussed. The saturation and value (SV) combination has also shown similar high performance using non-pyramidal processing as it extracts the shades from the image instead of Hue. Whereas, the individual Value component using non-pyramidal approach has shown a lower performance in the absence of the local contrast extraction technique of the pyramidal approach. This is because it tries to extract the brighter region from the image which may or may not be salient.

The overall feature extraction using non-pyramidal performance is better than the pyramidal processing. It is due to the fact that Itti attention model [14] extracts local contrast from the features for fixation. The local contrast extraction process is similar to the human perception. In the non-pyramidal approach, the overall features of the image are considered and does not fixate on one point but on the whole object. That is why the non-pyramidal approach is better when detection of the whole object is required. Whereas the performance of the pyramidal approach depends on the content of the image. If the object to be detected is small then the whole object will be considered for fixation. If the object to be detected is covering most of the image then only some parts of the object will be considered as salient. In this case, the non-pyramidal approach detects object better than the pyramidal approach.

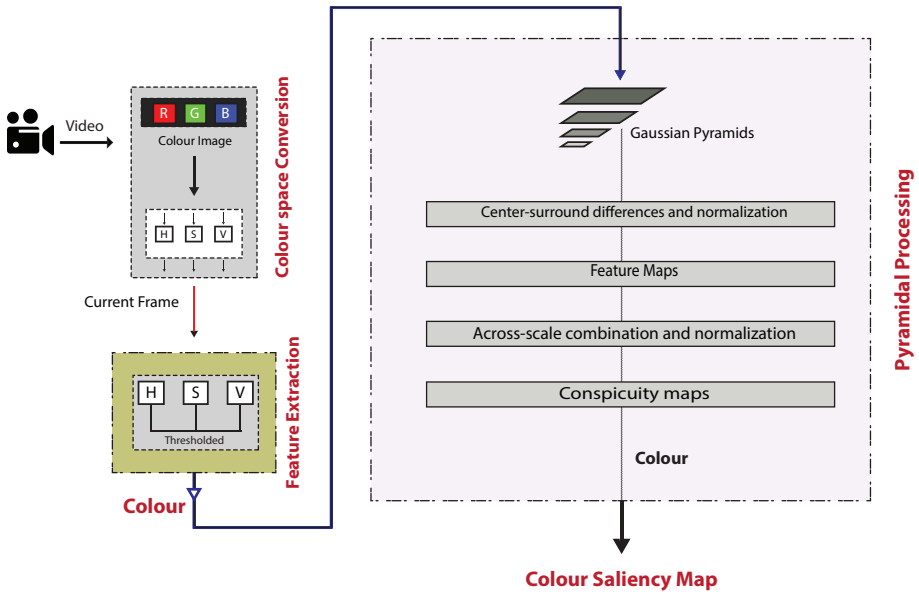


Figure 4.11: The modified version of colour used in the proposed MSD model. (this is a subset of Figure 4.1).

4.3 Colour feature computation

The previous experiments have demonstrated the effectiveness of the HSV colour space in detecting general salient features using pyramidal or non-pyramidal approach. However, in many tracking tasks we are interested in finding known targets such as those of a particular colour. We, therefore, would like to know which colour space is most effective in finding targets of known colour. We have used the colour feature to measure the performance of the system to find a particular coloured object. The subset of Figure 4.1 is shown in the model diagram of colour feature in Figure 4.11. As shown in Figure 4.11, the current frame is converted from RGB to HSV. After this conversion, the colour feature is extracted using thresholding. Thresholding is a segmentation method to divide the image into some pre-specified colour values to detect specific targets from a frame. Here, colour based thresholding is done using the known coloured values of the targets for creating a binary mask for each of the components of the

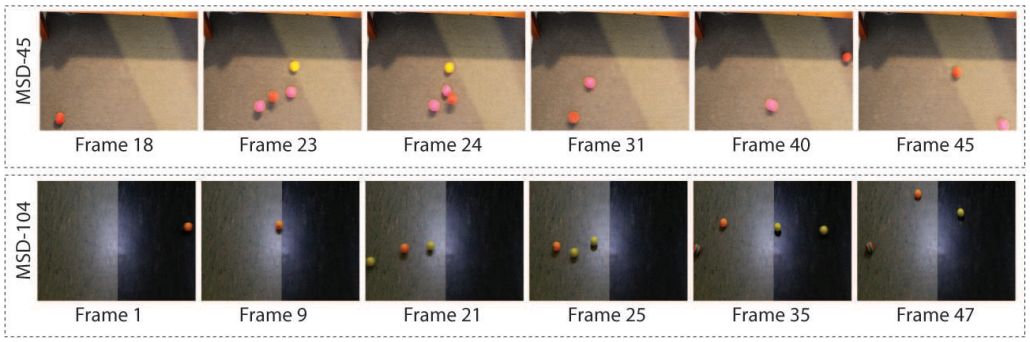


Figure 4.12: Sample images (a) MSD-45 (b) MSD-47.

colour space, as explained in Section 4.3.2. The colour feature extraction is followed by the Itti pyramidal approach to generate the saliency map based on the colour feature.

4.3.1 Experimental Evaluation

To evaluate and generate the saliency maps for the colour feature, we have created two datasets as follows:

1. **MSD-45 Dataset**
   As shown in Figure 4.12, the MSD-45 dataset has 45 frames with a different number of moving objects. These objects are of various colour with one red object in each frame. We have used this dataset to evaluate the colour feature to detect a target. The ground truth of this dataset is drawn by using a binary mask on the red object present in each frame.

2. **MSD-104 Dataset**
   The second dataset has an orange object in each frame along with other coloured moving objects. The ground truth is created by drawing a binary mask on the orange coloured objects present in each frame. The background is simple having some amount of illumination in the middle in every frame.

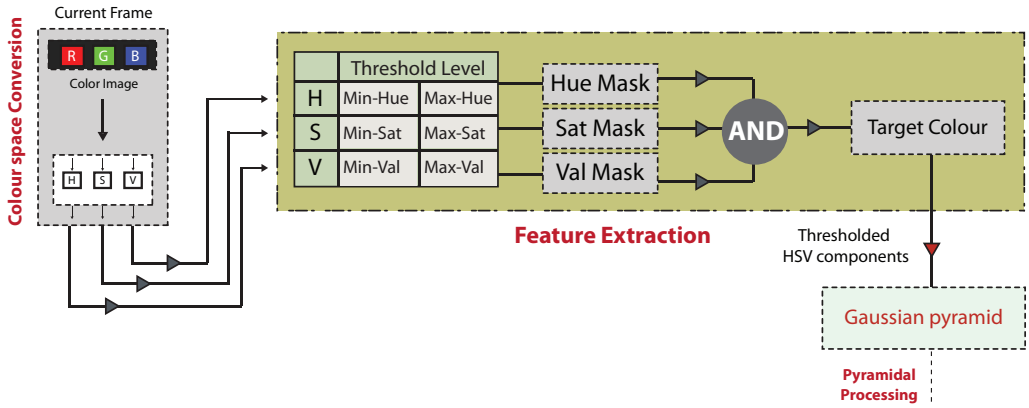


Figure 4.13: Colour thresholding technique used in our system. (this illustrate the working of the feature extraction box of Figure 4.11.)

We have compared the Itti model using the RGB and HSV colourspaces to detect a particular coloured target, as shown in Figure 4.12. To find the specific coloured target, we have thresholded the RGB and HSV colour space separately. The detail of the thresholding process is explained in the next section.

4.3.2 Colour Thresholding

Colour thresholding is a segmentation technique to partition an image according to the predefined colour values. We used the predefined colour values to specify a mask to segment each component of the colour space separately. These masks are then combined using the logical AND operation. The AND operation keeps the common pixels from the foreground if all three components of the colour space lie within the selected threshold. Figure 4.13 illustrates the working of the feature extraction box as shown in Figure 4.11. The colour space of the current frame is converted from RGB to HSV. The components of HSV colour space are then thresholded to extract the target colour. The thresholded HSV components used to create the Gaussian pyramids using the Itti model.

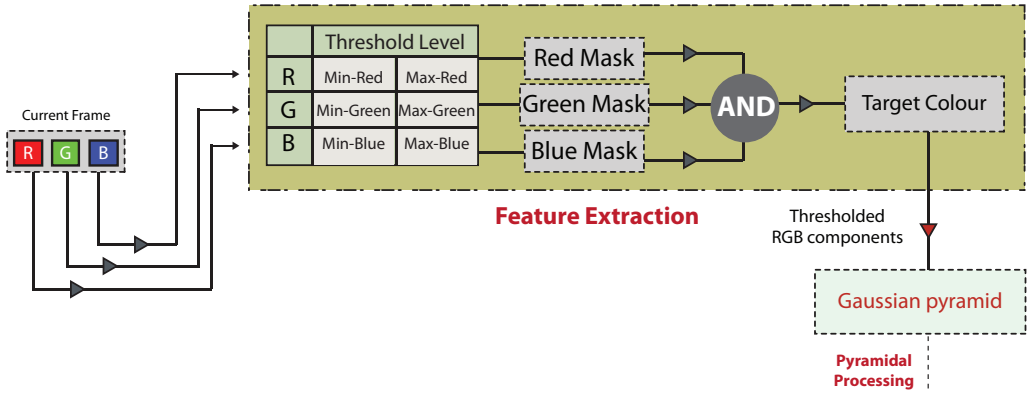


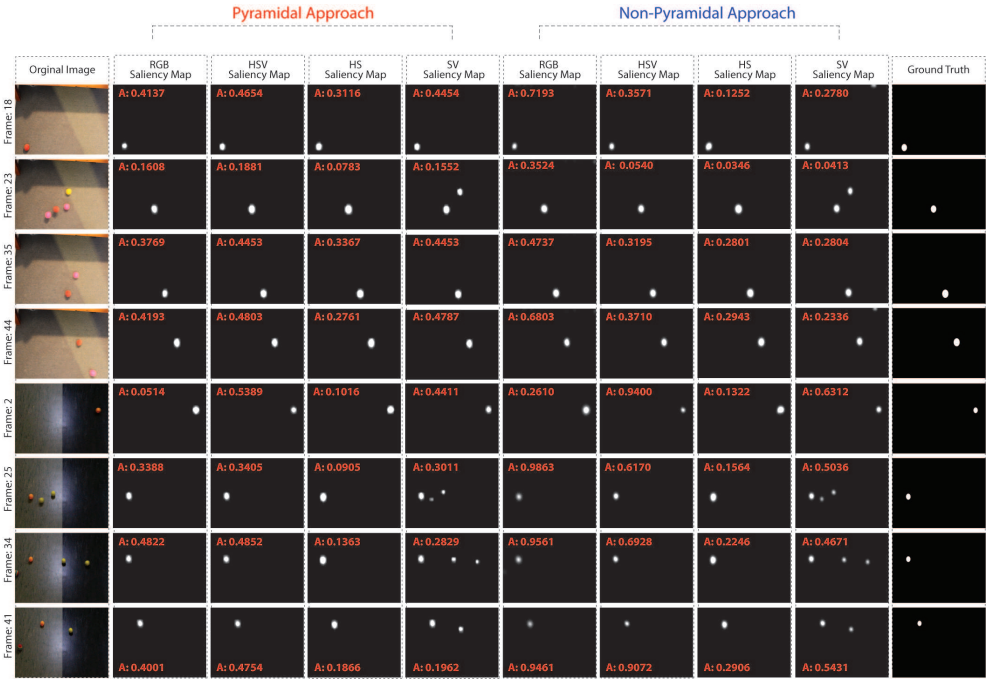
Figure 4.14: Colour thresholding technique (RGB colour space) used to compare Itti model with thresholded HSV colour space as shown in Figure 4.13.

We have thresholded RGB colour space to compare with the performance of the system with thresholded HSV colour space. As shown in Figure 4.14, RGB components are designated to specific colour values to extract the target colour. The minimum and maximum threshold levels are defined using a colour thresholder application of Matlab software. These levels are used to create the mask for each component. These components are combined using the AND operations to keep the common pixels to define the target colour.

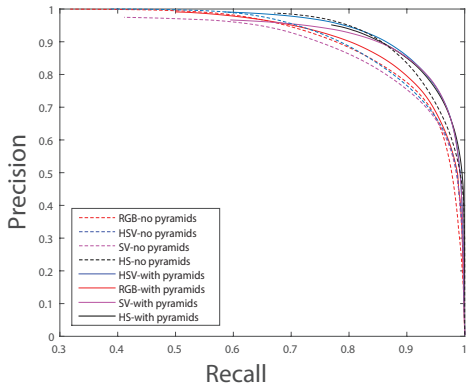
4.3.3 Target Detection Results

We have used the MSD-45 and MSD-104 dataset to analyse the performance of the colour thresholding method using the RGB and HSV colour space. The frames from these datasets are thresholded, as explained in the previous section. For the MSD-45, we have used red colour for thresholding whereas for the MSD-47 orange colour values are used as threshold level. The RGB and HSV colour space used these threshold level to extract the target colour.

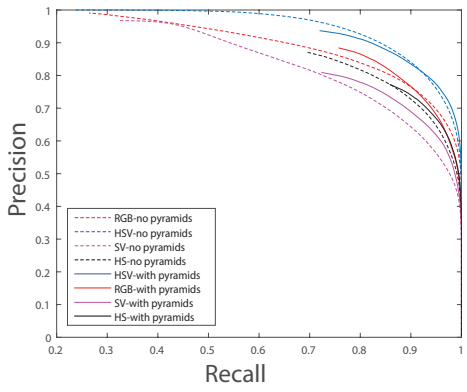
Some sample results are shown in Figure 4.15. The HSV saliency maps



(a)



(b) MSD-45



(c) MSD-104

Figure 4.15: Result images (MSD-45 and MSD-104 datasets) using the pyramidal and non-pyramidal approach (a). (b)Precision-Recall performance evaluation.

give better performance as compared to RGB in term of detecting coloured objects from the original images in both of the cases. It is because HSV

components decouple the Hue from the intensity and can be tuned to identify specific target feature quickly.

The thresholded Hue-Saturation combination also shows a good performance as this combination considered the Hue and shades to detect the specific colour from an image. Whereas the thresholded Saturation-Value combination gives a low performance as here, saturation tends to pop-out the colour close to the target colour that results in the detection of other similar shaded objects. So for pure colour segmentation using the HSV colour space is better than Saturation-Value as Hue gives the information about target colour values.

The non-pyramidal and pyramidal version has shown similar performance while detecting the target with known colour as shown in Figure 4.15. For MSD-45, both of the approaches show nearly equal performance whereas the non-pyramidal approach has shown high Recall values for the MSD-47 dataset.

4.4 Motion feature Computation

Previously, we have used the colour feature to identify the specific coloured objects from a scene. The colour feature is computed using a single frame separately, which is beneficial when a particular target needs to be identified. There is no information about the change of colour information between two consecutive frames. If this change of colour information is computed between two frames than motion can be calculated to detect all the moving object from a scene. The motion feature is an important feature that can detect changes in the position of an object relative to its background. The colour feature as discussed in Section 4.2.3 and the motion feature can be combined to identify a specific coloured target that is moving from one frame to another. We have implemented the motion feature using several sub-features such as colour change, optical flow analysis and background subtraction. The detail implementation of the motion

feature using sub-features is as follows:

4.4.1 Colour Change

The colour change feature plays an important role in identifying moving objects in a scenario. Here, we have used the temporal differencing method [83] to identify this colour change between consecutive frames in a video sequence. Temporal differencing is used to detect moving objects by taking the difference of consecutive frames $t-1$ and t. We computed this feature using HSV colourmap with pyramidal and non-pyramidal approaches for motion detection which is as follows:

4.4.1.1 Colour change analysis using the pyramidal approach.

Itti et al. [14] have used a pyramidal approach to detect a salient object from an image using the local contrast of the difference between a region and its surroundings, as discussed in Section 4.2.4. To compute the colour change information between two consecutive frames in a video sequence, we have compared two approaches using pyramids that are as follows:

1. In the first approach, as shown in Figure 4.16(a), we have computed the difference between the current and previous video frames using the temporal differencing method. The resulted frame is then further process by the pyramidal approach to detect salient objects. Here, nine spatial scales using Gaussian pyramids are formed consisting of the low-pass filtered versions of the resulted frame. The resulted frame is sub-sampled by a factor of 2 at each stage. The input frames are converted from RGB colour space to HSV space using the same procedure as described in Section 4.2.

2. In the second approach, the first nine spatial scales using dyadic Gaussian pyramids are created consist of a low-pass filtered version of the input image. This version is sub-sampled by a factor of 2

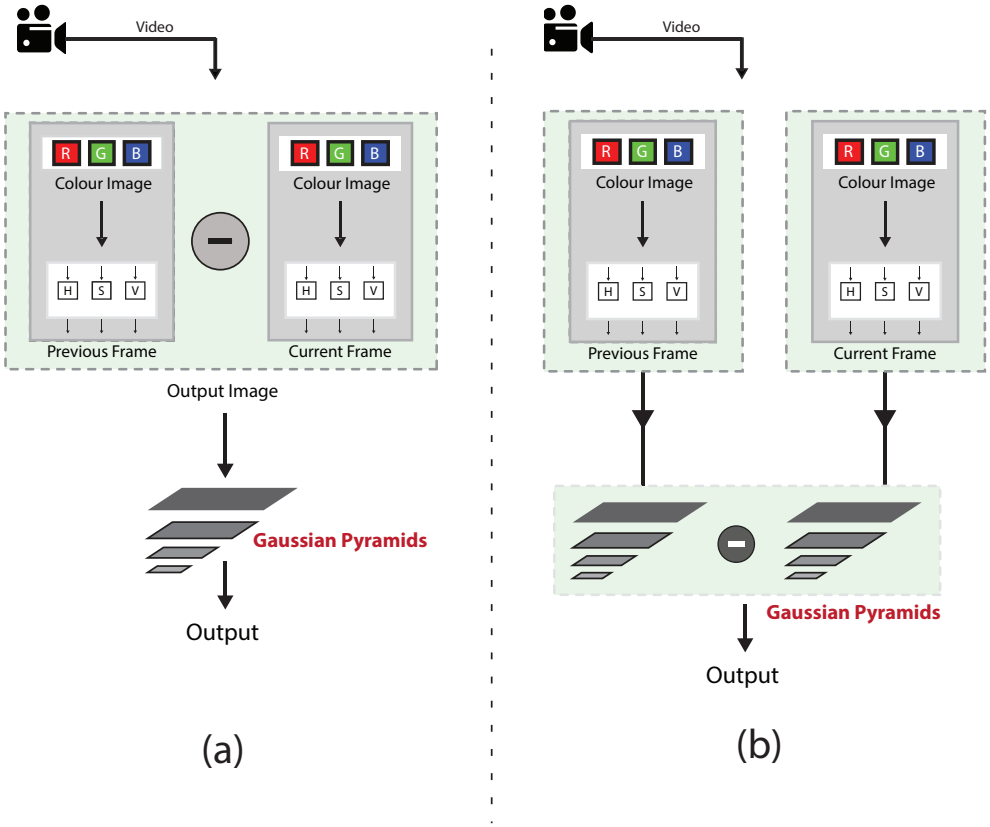


Figure 4.16: Colour change feature to detect motion (this is a subset of Figure 4.1) using the pyramidal approach (a) First approach (b) Second approach.

at each stage and represent the smaller objects. These nine spatial scales of the current and previous frames are subtracted to compute the colour change feature between consecutive frames, as shown in Figure 4.16(b).

When comparing these two temporal differencing methods using the pyramidal approach, we got the same results as shown in Figure 4.17. It does not make a difference if we subtract the consecutive frames before or during the pyramidal processing. Therefore, we can choose either method of temporal differencing when using pyramidal processing.

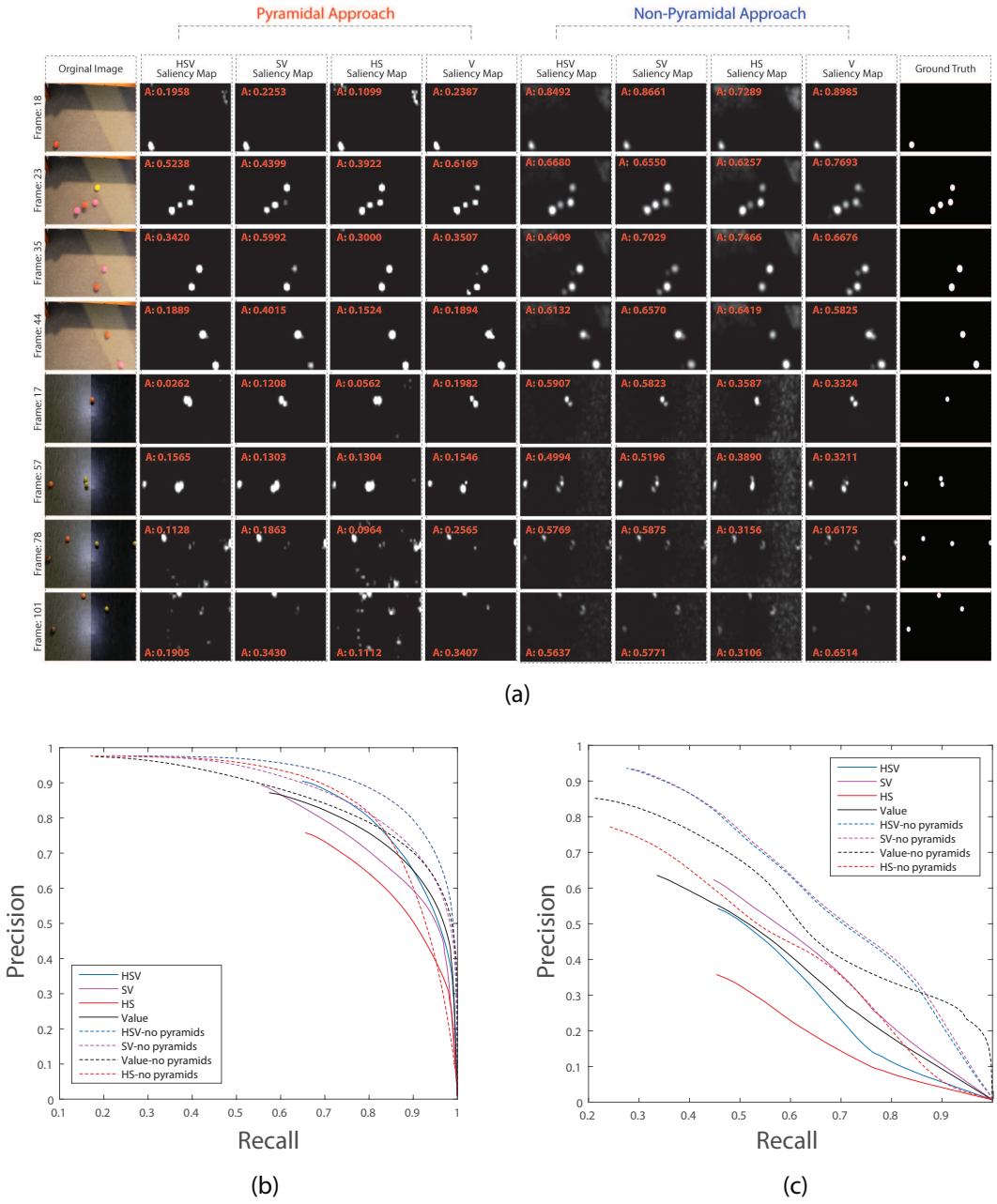


Figure 4.17: Sample images (MSD-45 and MSD-104 datasets) (a) using pyramidal and non-pyramidal approach. (b) Precision-Recall performance evaluation

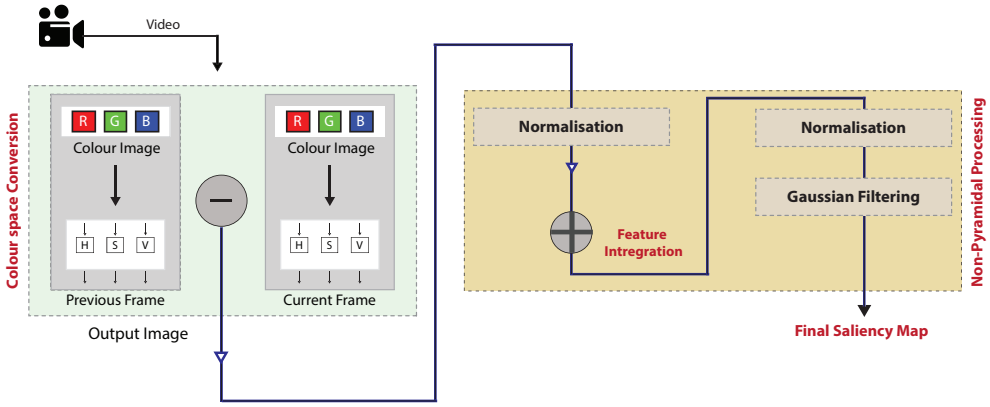


Figure 4.18: Colour change feature using the non-pyramidal approach.

4.4.1.2 Colour change analysis using the non-pyramidal approach.

In the non-pyramidal approach, colour change feature is computed by subtracting the current and previous frames. Figure 4.18 shows the simple model in which the RGB colour space is converted to HSV colour space for each frame. Each channel of the current frame is subtracted from the corresponding channel of the previous frame. The output image after the temporal differencing is processed by the non-pyramidal approach as discussed in Section 4.2.4.

We have used the MSD-45 and MSD-104 datasets as shown in Figure 4.12, for the analysis of colour change feature using pyramidal or non-pyramidal processing. The ground truth of these two datasets is drawn again using the binary mask to mark only the moving objects present in these datasets.

The sample images and the Precision-Recall curve, as shown in Figure 4.17 show the performance of the moving object detection using pyramidal and non-pyramidal approachs. It can be seen that HSV colour space, when used with all components, is highly adaptive in detecting salient moving objects. The Precision-Recall score of the non-pyramidal approach is higher than for the pyramidal approach. The HSV-based pyramidal processing show performance as temporal differencing using the HSV colour

space detected some of the trailing regions, also called as *ghost regions*. These regions become prominent using the centre-surround differencing at various scales. Whereas in non-pyramidal processing, these *ghost regions* are blurred using the Gaussian filtering and standard normalisation. The individual Value and combination of Value component with Saturation component have also performed effectively using the non-pyramidal approach because the Value component helps to extract the brighter regions while detecting different shades of the moving objects. Overall, the non-pyramidal approach has shown high performance while using the temporal differencing method for moving object detection.

4.4.2 Optical Flow

Feature selection is a critical part of any motion saliency system as it defines the uniqueness of a moving object within its surroundings. Optical flow is one promising estimation technique that determines the displacement of the pixels between frames of a video sequence [58]. This method can detect dense correspondence fields even from a moving camera and estimate the flow of the field by minimising the brightness of the corresponding pixels of a scenario [99].

We have incorporated the layer based method as described in [57] and used optical flow as a saliency detection feature to evaluate the performance of our system.

4.4.2.1 Experimental Evaluation

Here, the flow of the object's motion is estimated between consecutive frames. Therefore, we conducted experiments on the following datasets, as shown in Figure 4.19, to examine the optical flow feature using the pyramidal and non-pyramidal approachs.

1. **MSD-31 Dataset**
   This dataset is a collection of yellow coloured objects in every frame

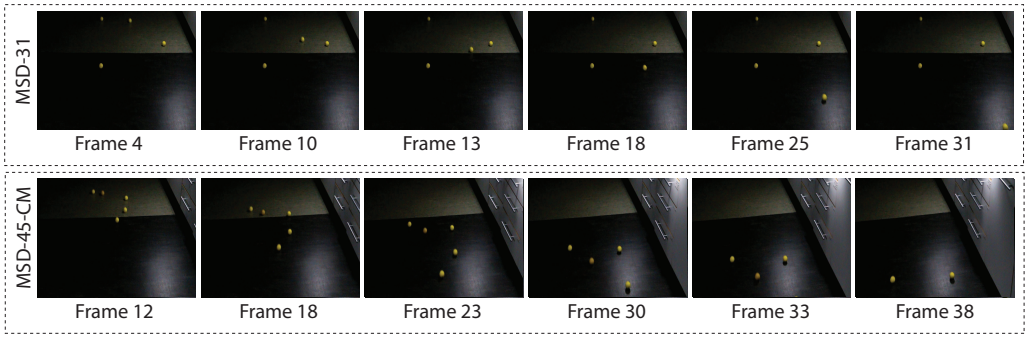


Figure 4.19: Sample images (a) MSD-31 (b) MSD-45-CM.

in which only one is moving. The background is dark and having the small patch of brightness at one of the corners.

2. **MSD-45-CM Dataset**
   The last dataset is a challenging dataset in which the moving camera scenario is presented. It consists of all objects that are moving along with the camera.

The details of the analysis are described as follows:

4.4.2.2 Optical Flow analysis using Pyramidal approach

Here, we would like to investigate whether optical flow can be integrated with our system using the pyramidal approach. Previously, we analysed the colour (to detect specifically coloured targets) and colour change (to detect the moving objects) features. The optical flow is used as the second motion feature to detect salient moving objects based on the layer-based method as described in [57] to examine some challenging datasets with moving the camera or background scenarios. Figure 4.20 shows simple model diagram of optical flow (the subset of Figure 4.1).

Consecutive frames such as the previous and current frames are used for the estimation of the optical flow. Then the layer-wise optical flow is estimated using a mask that indicates the visibility of each layer. An iterative reweighted least squares (IRLS) method [57] is used for optimisation

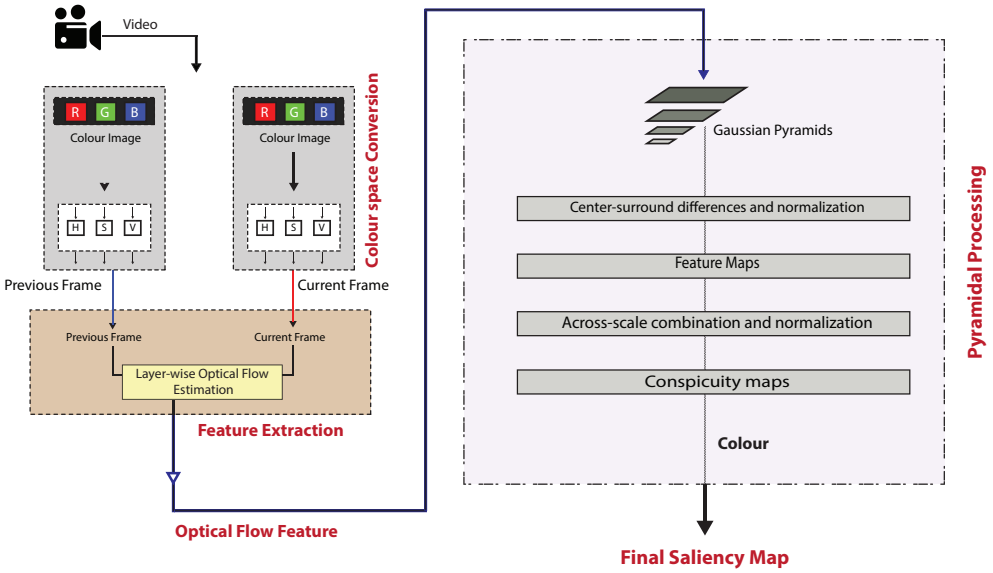


Figure 4.20: Optical flow estimation using the pyramidal approach (this is a subset of Figure 4.1).

that is followed by flow computation at each pyramid level. The layer mask is updated using the estimated flow.

The parameters defined in the equation as described in Section 2.11 (on page 2.11) can be adjusted to produce a smoother flow field. These parameters include α that is a regularisation weight, the ratio for downsampling the image, the width of the coarsest level of the pyramid and the number of outer and inner fixed point iterations for computing IRLS used in optical flow algorithm as described in [90].

The coarse to fine levels are computed using consecutive frames to yield the flow field in the x and y directions and the wrapping image. For the computation of optical flow, the image intensity at pixel location x, y at time t is denoted by $I(x, y, t)$, where I is a current frame. The associated flow vector is represented by the velocity magnitude $r_{x,y,t} \geq 0$:

$$r_{x,y,t} = \sqrt{V_x^2 + V_y^2} \tag{4.12}$$

Where V_x and V_y are, respectively, the horizontal and vertical optical

flow components. The salient regions are detected by creating the Gaussian pyramids of the flow vector r represented the directions for each frame. The final saliency maps are computed by using the local contrast of the difference between a region and its surroundings, as discussed in Section 4.2.4.

We have not compared the performance of optical flow feature with the ground truth of the dataset, as it requires special motion annotated ground truth for comparison. The creation of such ground truth is time-consuming. Different work is done for annotating motion for creating ground truth of real-world videos [57, 100, 101]. For our work, we require the final motion saliency map after the integration of different features, so we leave the comparison of optical flow results with the motion annotated ground truth for future work.

In Figure 4.21, we demonstrate sample images from the analysis of using optical flow as a saliency feature. It can be seen that the search space is reduced by locating the region with dense flow between consecutive frames. Figure 4.21 shows one of the frames from each dataset along with its ground truth. The ground truth is based on the binary mask of the actual position of the objects present in the original image.

The pyramidal approach of Itti et al. [14] detects the irregular velocity magnitude of the moving object. We can visualise this detection by comparing the flow results of the three datasets as shown in Figure 4.21. Here, the detection using optical flow is performed in a bottom-up manner without giving any target information.

The three datasets show different challenging situations. In the first dataset, MSD-45, some of the object occluded in some of the frames or come close to each other. This effect can be seen from the sample image shown in the results where the flow of overall moving objects are considered and making it difficult to detect the final objects using the optical flow. In the second dataset MSD-31, there is one moving object present that is easily detected by the saliency method. Optical flow methods can

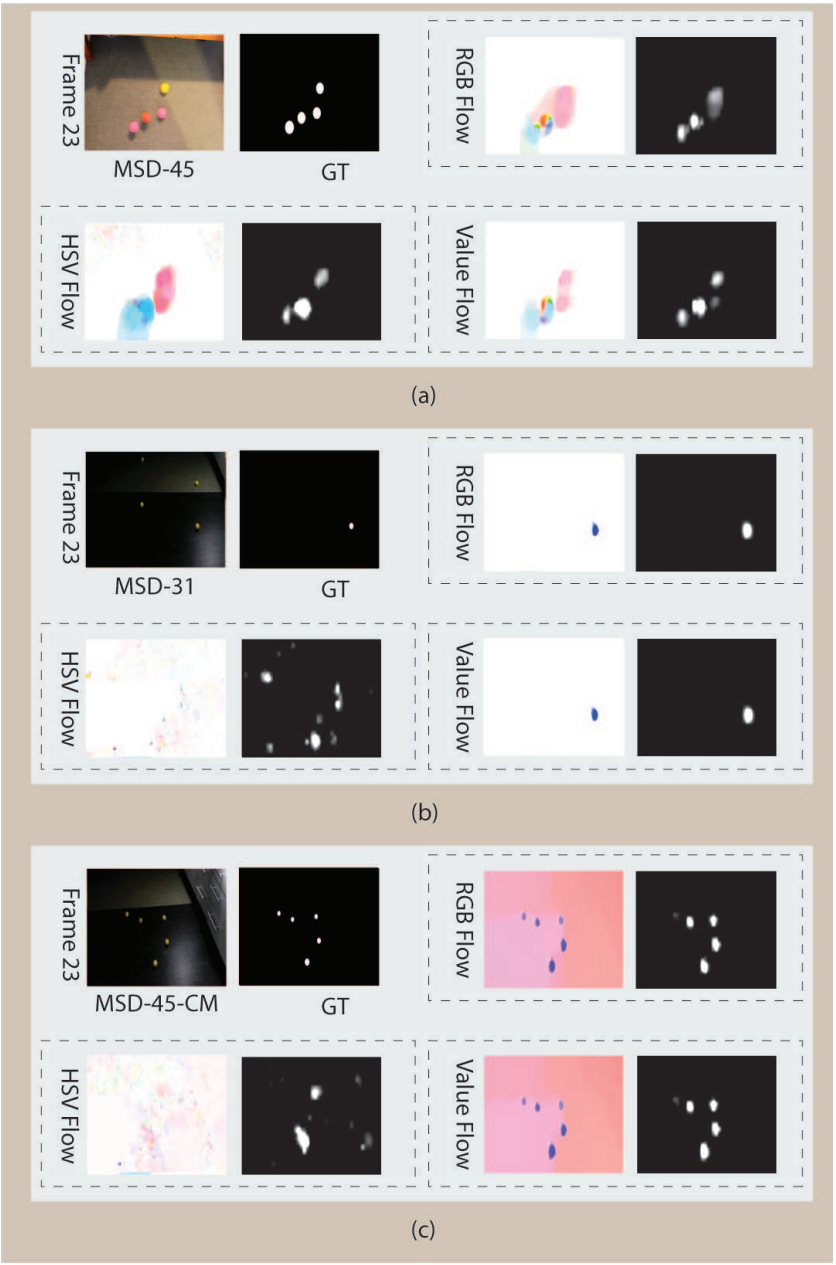


Figure 4.21: Sample images from the Optical Flow feature computation using the pyramidal approach (a) MSD-45 (b) MSD-31 (c) MSD-45-CM. Optical flow estimation is done using the RGB and HSV colour space.

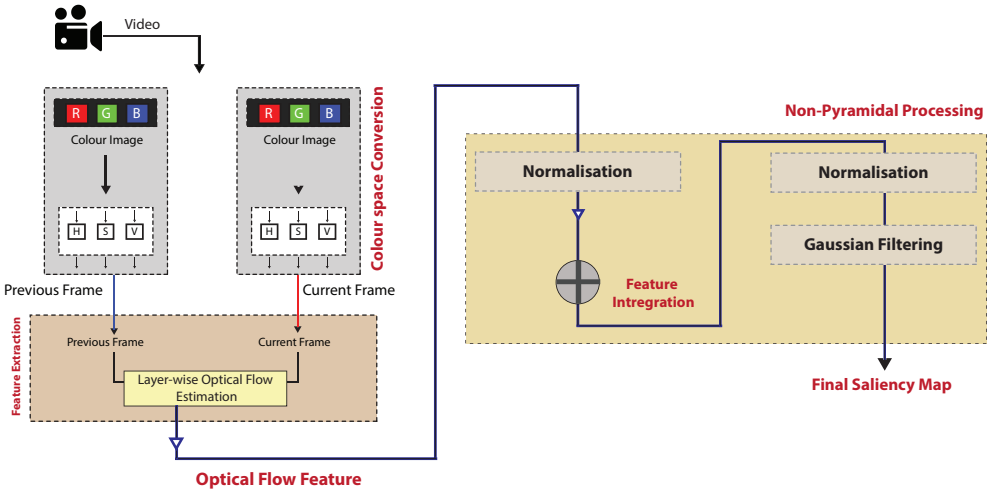


Figure 4.22: Optical flow estimation using the non pyramidal approach (this is a subset of Figure 4.1).

identify the flow of moving objects in moving camera situations. The last dataset MSD-45-CM, represents the moving camera situation. The results of this dataset show that the pyramidal approach can identify the object by considering the flow between the consecutive frames.

Here, the individual Value component flow is much better than the RGB and HSV flow in which all channels are considered. It is because Hue and Saturation tend to detect different shades from the image, whereas the individual Value component detects the brightness change from one frame to another. As we have already seen that RGB has a tight correlation among the components, it also identifies the intensity change using optical flow, but it is illumination dependent. In some of the frames, the RGB colour space shows low performance while merging different object flow together, as shown in Figure 4.21(a). While the individual value component improves detection by detecting the salient moving object flow.

4.4.2.3 Optical Flow Feature using the non-pyramidal Approach

Here, the optical flow using the non-pyramidal approach is examined to detect salient objects. From the previous analysis of optical flow using the pyramidal approach, we can see that optical flow efficiently estimates the flow of the motion of the objects. The individual Value components help to detect the brighter salient moving objects between consecutive frames. Here, we used the layer-based method to estimate optical flow described in [57] without local contrast information. The model diagram (a subset of Figure 4.1) shows the working of the Optical flow, as shown in Figure 4.22.

The consecutive frames, as shown in Figure 4.22 are converted from RGB to HSV colour space. Optical flow is estimated using the same layer-based technique as discussed in [57]. After computing the flow between the consecutive frames, optical flow result is normalised to suppress the unwanted noise from the estimation. This normalised optical flow result is followed by the same non-pyramidal procedure as discussed in Section 4.2.4.

The sample images from the analysis of the optical flow using the non-pyramidal processing is shown in Figure 4.23. We have used three datasets (MSD-45, MSD-31 and MSD-45-CM) to compute the flow between the consecutive frames. Here, we have not performed the performance analysis of the optical flow result with the ground truth of the dataset. As we have discussed in the previous section that computing the ground truth to analyse optical flow itself is a challenging task. This computation of the motion annotated ground truth is out of the scope of this work.

Here, we are interested in computing optical flow to integrate it with other features to produce the final saliency map. The final saliency map is compared to evaluate the effective of the proposed system later in this chapter. For this work, the optical flow performance can be visualised by comparing the irregular velocity magnitudes with the original frames.

From the sample images shown in Figure 4.23, it can be seen that the non-pyramidal processing has shown low performance by computing the

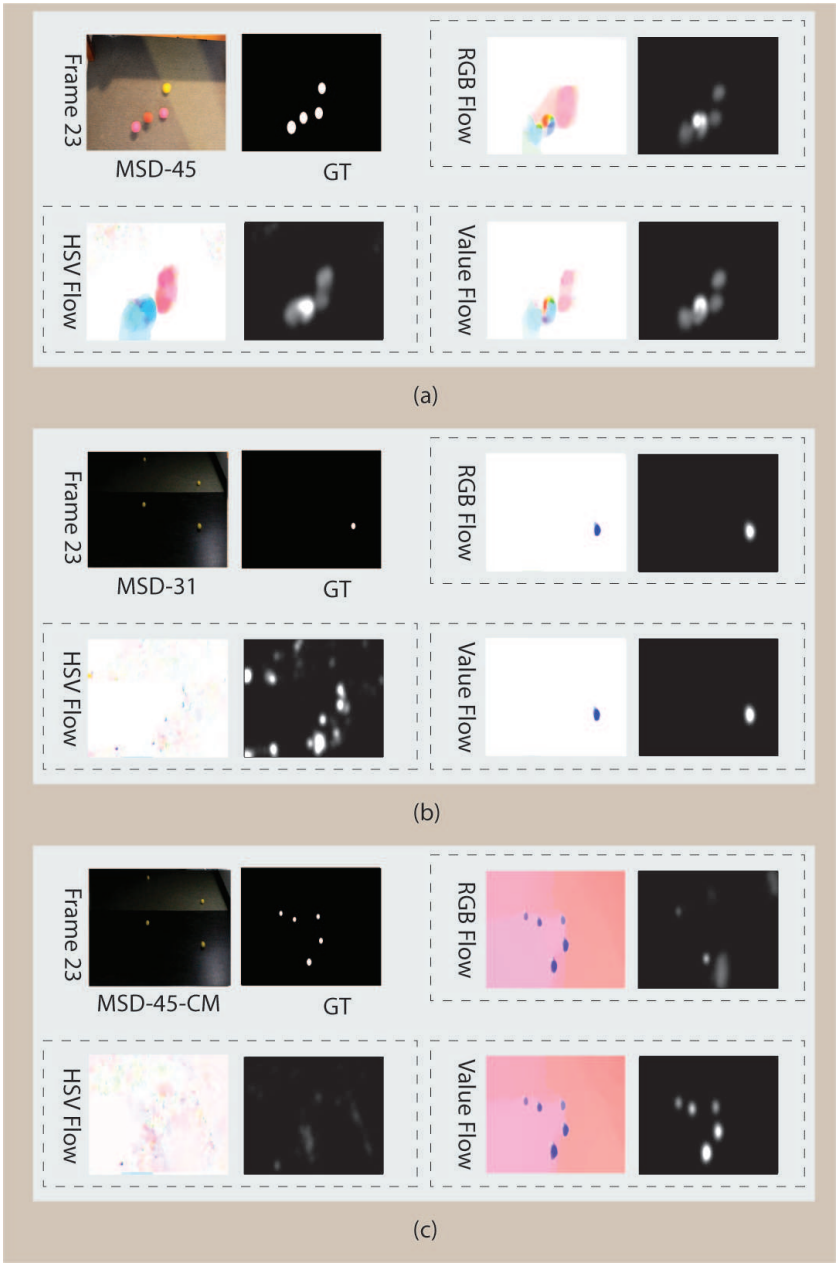


Figure 4.23: Sample images from the optical Flow feature computation using Non-pyramidal approach (a) MSD-45 (b) MSD-31 (c) MSD-45-CM. Optical flow estimation is done using the RGB and HSV colour space.

saliency map using the optical flow. In the absence of the center-surround local contrast, the salient objects were detected from the Optical flow result using simple normalisation and Gaussian filtering.

Here, the RGB and HSV colour spaces have shown low performance than the individual Value component of the HSV colour space. This is due to the fact that the Value component extracts the brighter regions from the flow result. The RGB colour space shows low performance when datasets are challenging (MSD-45 and MSD-45-CM). We can see from the first sample image that the objects are not detected using the RGB colour space followed by non-pyramidal processing. The third image forms the moving camera dataset, the non-pyramidal processing is not effective enough to detect the objects using the RGB colour space. The same is true for HSV colour space when all channels are used. The optical flow computed using the HSV colour space is not efficient due to the inclusion of Hue and Saturation that detect shades from the background along with the objects.

To summarise the results, the overall optical feature extraction using the pyramidal processing is better than the non-pyramidal processing. This is due to the fact that feature fixation is performed in Itti et al. [14] using local contrast as discussed in the previous sections. The individual Value component is better than the RGB or HSV when all components are used. We have, therefore, chosen the individual Value components for computing the optical flow for further analysis that is presented later in this chapter.

4.4.3 Background Subtraction

Background subtraction is another promising technique to identify moving objects in a scenario. Moving objects can be detected using a reference background image of a scenario called *Background*. This reference background image can be estimated when there is no moving object in the scenario. Significant deviations from this reference image can be computed

to detect moving objects in each frame [5]. Modelling a reference image is a challenging task as it requires different situations to be considered, such as illumination changes, shadow removal, and occlusion.

Here, we are interested to use a simple background subtraction technique to remove non-salient locations from the saliency map perhaps apparently salient background areas as explained in Section 2.12 (on page 2.12). Here we would like to determine if there is a significant performance increase in the object detection result after adding this feature.

For a simple test, we have considered a simple background image of a scenario captured during the creation of a dataset. The background image is updated using Equation 2.43. Then a predefined threshold that is found empirically to mark a pixel at location (x, y) as a foreground using Equation 2.42 explained in Section 2.12 (on page 2.12).

4.4.3.1 Experimental Evaluation

We have tested background subtraction using the following datasets as shown in Figure 4.24. The result of this analysis is compared with the ground truth of the datasets.

1. **MSD-11** This dataset is having 11 frames in which moving objects are shown. We have selected a background image from the same scenario without any moving objects to perform background subtraction.

2. **MSD-104** This dataset has 104 frames that are subtracted using the background frame to extract the background subtraction feature.

3. **MSD-82** In this dataset, there are 82 frames with static and moving objects. Some static objects start moving when other moving objects occlude with them. We selected this dataset to check if we can detect these objects that start moving after occlusion in the middle of the scenario. We have manually selected one of the images as the background image.

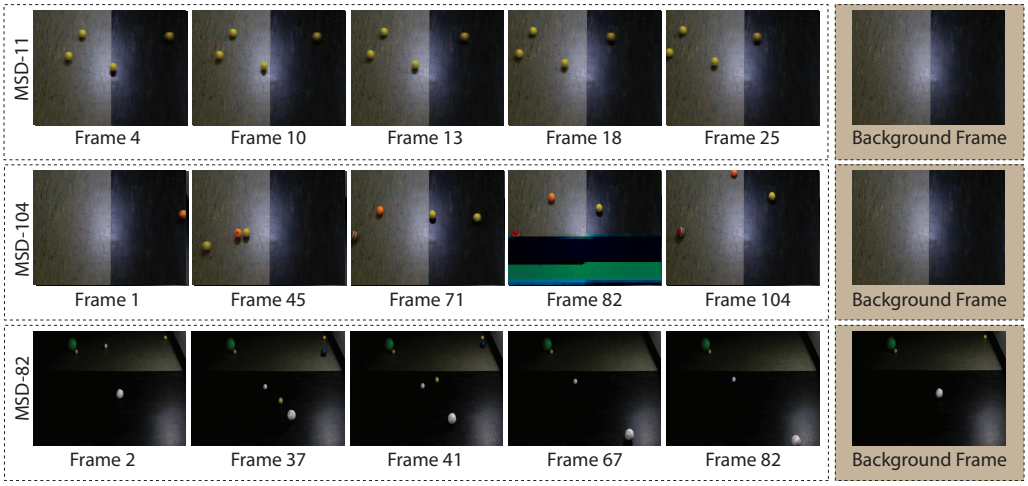


Figure 4.24: Sample frames from the datasets to analyse the Background Subtraction feature. Background frame of each of the three datasets (MSD-11, msd-104 and MSD-82) is also shown.

The detailed background subtraction analysis is as follow:

4.4.3.2 Background Subtraction using Pyramidal approach

Here, an adaptive background subtraction technique is used with the pyramidal approach based on Itti attention model [14]. Here, pyramidal processing identifies the salient regions from the scenario while background subtraction helps in elimination of the illumination effect and non-salient regions from the saliency map. The background subtraction feature model diagram is shown in Figure 4.25. The initial background frame is selected with no moving objects. The first step is to convert the colour space of the current frame and the background frame from the RGB to HSV colour space. The next step to update the HSV based-background frame iteratively. Here, each component of the HSV is updated individually using the following equations:

$$\boldsymbol{H_{BG(t+1)}} = \alpha \boldsymbol{H_t} + (1-\alpha)\boldsymbol{H_{BG(t)}} \tag{4.13}$$

$$\boldsymbol{S_{BG(t+1)}} = \alpha \boldsymbol{S_t} + (1-\alpha)\boldsymbol{S_{BG(t)}} \tag{4.14}$$

$$\boldsymbol{V_{BG(t+1)}} = \alpha \boldsymbol{V_t} + (1-\alpha)\boldsymbol{V_{BG(t)}} \tag{4.15}$$

where α is 0.5 value that is found empirically in this experiment. The current frames $\boldsymbol{H_t}$, $\boldsymbol{S_t}$ and $\boldsymbol{V_t}$ are used to update the background slightly at frame t based on the HSV components such as $\boldsymbol{H_{BG(t)}}$, $\boldsymbol{S_{BG(t)}}$ and $\boldsymbol{V_{BG(t)}}$, depending on the α value. The updated backgrounds $\boldsymbol{H_{BG(t)}}$, $\boldsymbol{S_{BG(t)}}$, $\boldsymbol{V_{BG(t)}}$ are used for subtraction with the current frame based on HSV components using the following equation:

$$\boldsymbol{H_{Sub(t)}} = \big(\boldsymbol{H_t} - \boldsymbol{H_{BG(t+1)}}\big) > H_{TH} \tag{4.16}$$

$$\boldsymbol{S_{Sub(t)}} = \big(\boldsymbol{S_t} - \boldsymbol{S_{BG(t+1)}}\big) > S_{TH} \tag{4.17}$$

$$\boldsymbol{V_{Sub(t)}} = \big(\boldsymbol{V_t} - \boldsymbol{V_{BG(t+1)}}\big) > V_{TH} \tag{4.18}$$

Here, $\boldsymbol{H_{Sub(t)}}$, $\boldsymbol{S_{Sub(t)}}$ and $\boldsymbol{V_{Sub(t)}}$ are the background subtraction results based the individual HSV components. The HSV-based background subtraction result is thresholded after taking the difference with the current frame. We have selected an empirical threshold level to control the Hue, Saturation and Value information to select moving pixels of specific colour from the result. After thresholding, the results from the individual HSV components are combined using the AND operation. The thresholding process classifies the pixels as the foreground pixels using each component's result. The AND operation then eliminates unwanted pixels from the thresholded result by taking the common pixels from each component result.

A median filter is used to reduce noise and preserve the sharp edges from the detected pixels of the foreground. A different square neighbourhood size is found empirically for every dataset. The filtered result is fed

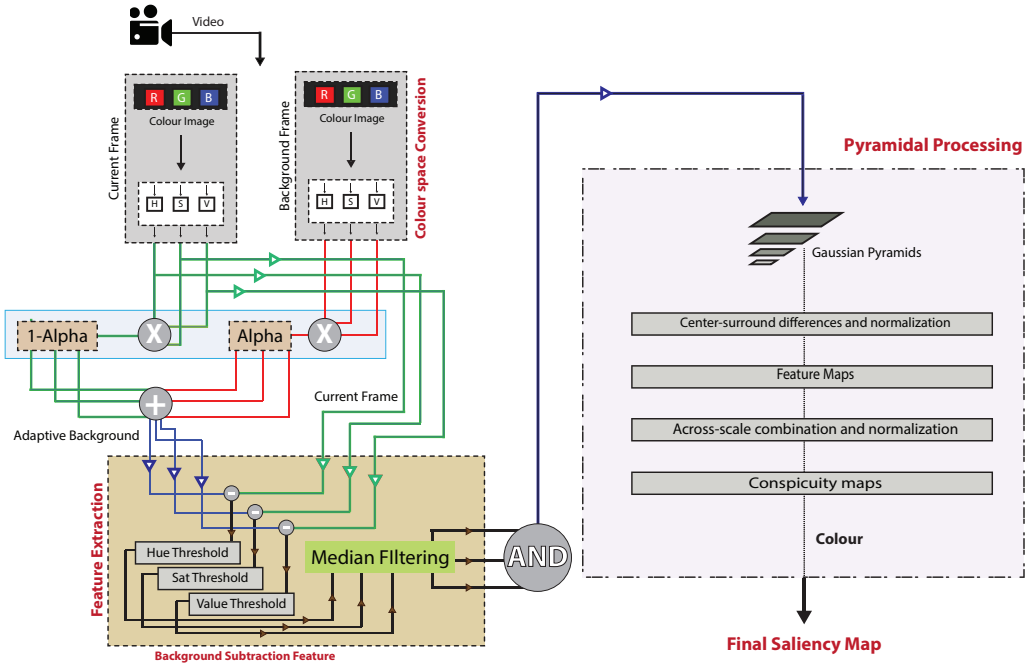


Figure 4.25: Background subtraction using the pyramidal approach.

into the pyramidal processing to extract local contrast. The complete process of process of pyramidal approach is explained in Section 4.2.4.

The sample images from the analysis of the background subtraction feature using the pyramidal approach are shown in Figure 4.25. The HSV-based saliency maps using all components of the HSV colour space show better performance than the individual Value component. It can be seen from Figure 4.25 frame 3 that all the moving objects are detected using all the components of the HSV colourspace whereas the individual Value missed some of the objects because when the brightness of the region is low then the Value component is not sufficient to detect the moving objects. When all the components of the HSV colour space are used then the information of the Hue and Saturation help in detecting the Hue and Shades even in the dark area of a frame.

The Precision-Recall curve for the pyramidal approach is shown in Figure 4.27. The HSV-based background subtraction using the pyramidal ap-

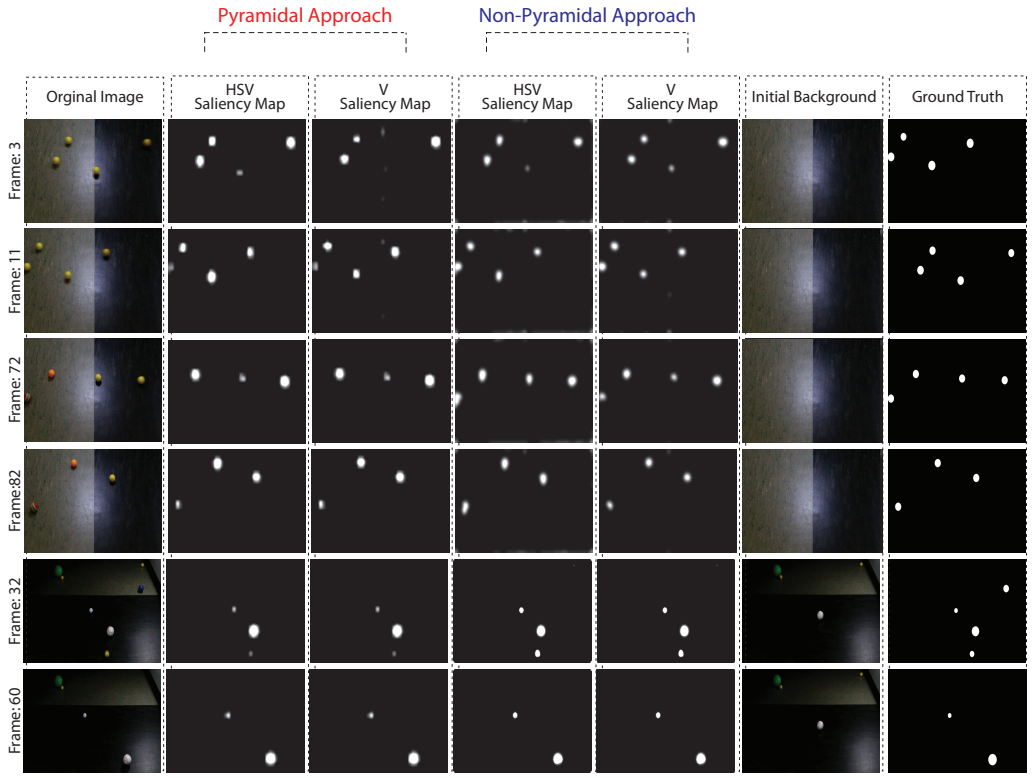


Figure 4.26: Sample images from the analysis of Background feature using pyramidal and non-pyramidal approaches (a) MSD-11 (b) MSD-104 (c) MSD-31.

proach has shown higher performance than individual Value component for all three datasets we have used.

4.4.3.3 Background Subtraction using Non-pyramidal approach

Here, we have analysed the background subtraction using non-pyramidal approach using the non-pyramidal approach as shown in Figure 4.28. It can be seen that normalisation is performed after combining the results of the thresholded HSV components. This normalisation process suppresses the unwanted noise from the background subtraction result. This normalised background subtraction result is followed by the same non-

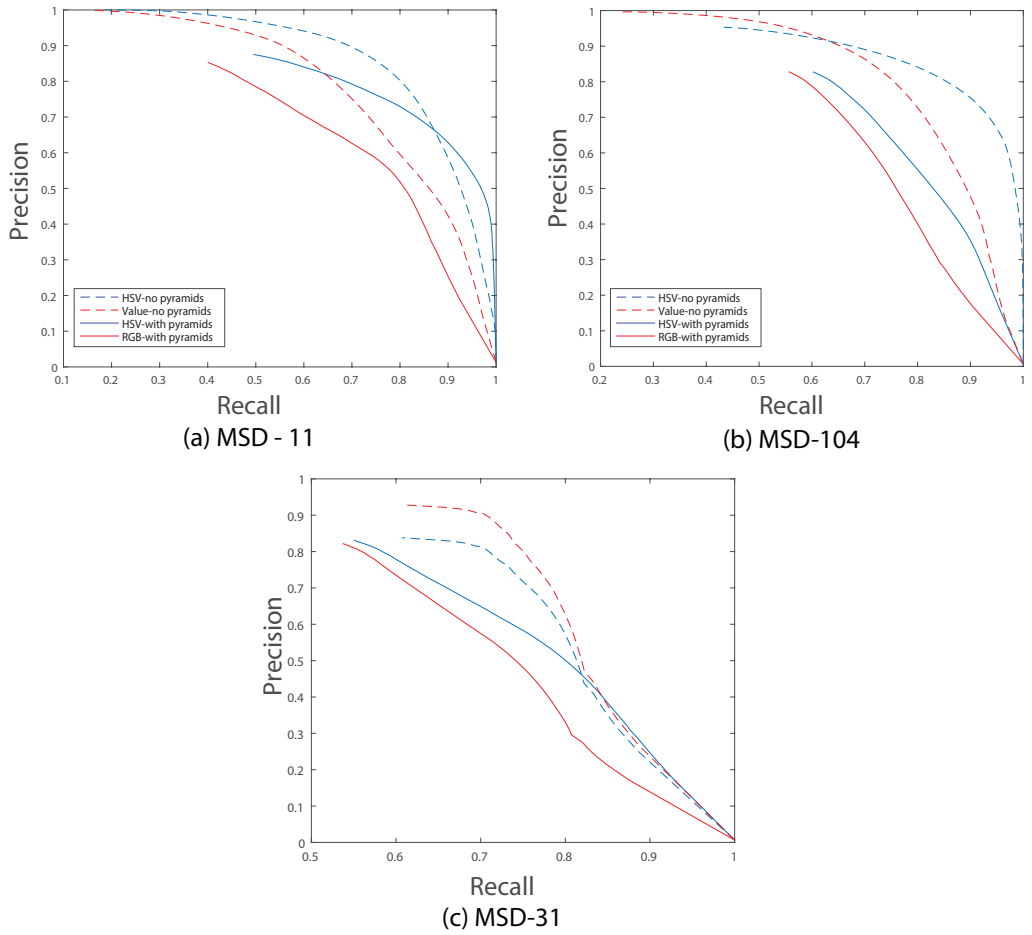


Figure 4.27: Performance measure of Background analysis using pyramidal and non-pyramidal approach in terms of Precision-recall curve.

pyramidal procedure as discussed in Section 4.2.4.

The performance of the background Subtraction feaute is shown in Figure 4.26. Here, the HSV-based non-pyramidal approach performed higher than the individual Value component. The HSV uses the Hue and Saturation components to extract the Hue and Saturation information of the object without finding the local contrast. The results from the Precision-Recall curve has shown that the background subtraction is effectively performed using the non-pyramidal processing as shown in Figure 4.27.

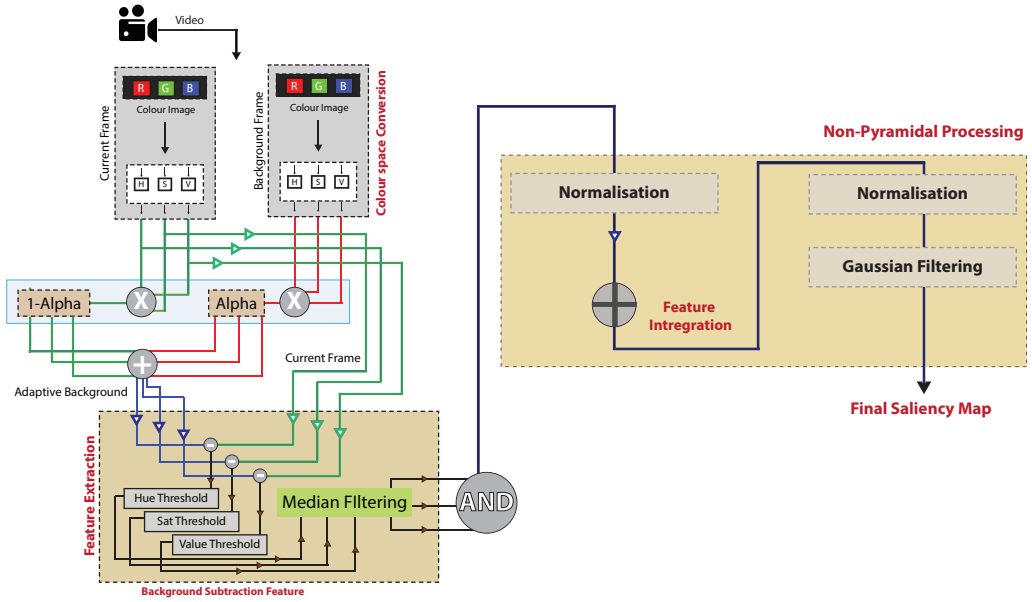


Figure 4.28: Background subtraction using the pyramidal approach.

The overall background subtraction feature extraction using non-pyramidal approach has various advantages over the pyramidal approach. Because there is no need to find the local contrast and the variation in the background and the current frames are automatically detected when subtracted together. Thus, we concluded that background subtraction is a promising feature for motion saliency using the HSV-based non-pyramidal approach that improves the salient object detection.

4.5 Feature integration

In this section, we would like to examine the choice of integrating the saliency maps using different features to form the final saliency map. Previously, we have analysed four new features such as colour, colour change, optical flow and background subtraction individually.

These four features detect moving objects with different level of efficiency. Here, we would like to analyse the combined effect of these fea-

tures to examine the performance of motion saliency detected. After examining the performance, we proposed a motion saliency method that will detect moving object efficiently. This motion saliency will be incorporated with the object tracking component to estimate the motion of the objects.

The feature integration is shown in the main model diagram Figure 4.1. From the figure, it can be seen that all the four features are combined after the pyramidal processing to produce the final saliency map. Integration is a mathematical operation in which different features are combined to produce the final saliency maps using element-wise multiplication, simple summation and logical operation.

In this motion saliency model, we have extracted the features to detect motion at different scales and combine these features to produce the feature maps in a centre-surround approach using Itti attention model [14]. These different saliency maps are combined together to generate a master saliency map. Here, we have examined two different feature combination methods that include arithmetic operations such as summation and multiplication. The detailed analysis of the feature integration are as follows:

Experimental Evaluation For the initial experiment, we have used summation and element-wise multiplication to combine different features. We have selected the MSD-82 datasets to evaluate the performance of feature integration. The results of the feature integration are compared with the ground truth using the Precision-Recall curve.

4.5.0.1 Feature Integration using the pyramidal approach

In this section, feature integration is explored using the pyramidal approach using Itti et al. model [14]. Feature extracted that is followed by the centre-surround mechanism. The centre-surround mechanism is performed using the cross-scale difference of Gaussian operation that generates feature maps for the four features such as colour, colour change, optical flow and background subtraction. These feature maps are then in-

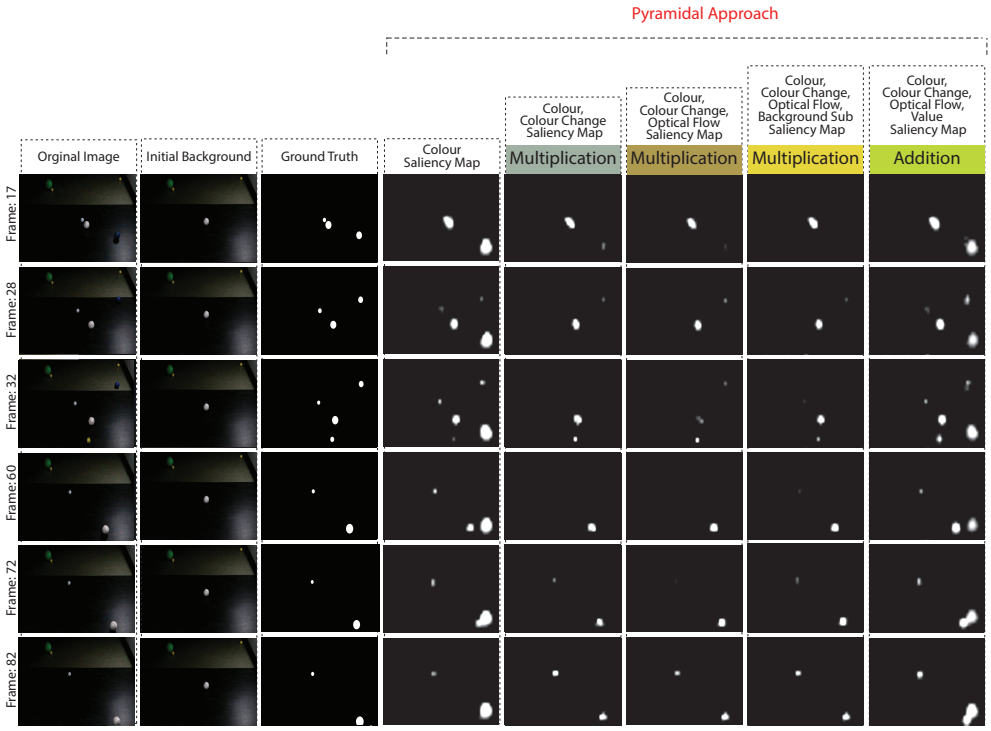


Figure 4.29: Sample images from the result of feature integration (Colour, Change Colour, Optical flow and Background Subtraction) using pyramidal approach.

tegrated and normalised at different scales to yield the two conspicuity maps. The final saliency map is then generated when these conspicuity maps are combined linearly with equal weights. The maximum of the saliency map defines the most salient image location.

We have used four features to detect motion in the MSD-82 dataset in which several objects are moving. Here, we have chosen to detect all the moving objects from this dataset. The colour feature based on the HSV colour space is thresholded for the values of all moving objects as we targeted different moving object throughout the scenario. The colour change feature based on the individual Value component detect all moving objects using the intensity information from the temporal difference. The

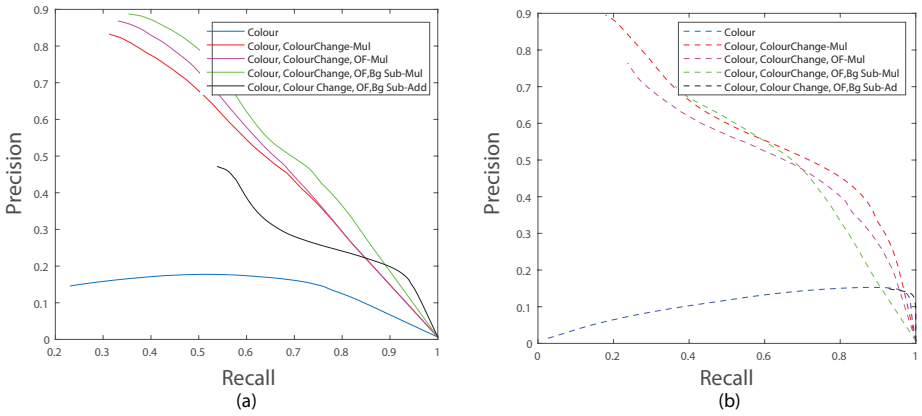


Figure 4.30: Performance measure of Background analysis using in terms of Precision-recall curve. (a)pyramidal approach (b) non-pyramidal approach

Optical flow estimates the flow between consecutive frames using the individual Value as it extracts the brighter regions from the flow result. The HSV-based background subtraction feature is used for detecting the moving object using the adaptive background method as we have discussed in the previous section.

4.5.0.2 Feature Integration using the non-pyramidal approach

For the final analysis, we have used non-pyramidal approach for integration. The process of non-pyramidal approach is discussed in Section 4.2.4. We have used the four feature to detect motion and combined the saliency maps generated by these feature maps using Arithmetic operations such as multiplication and addition. When all the features are integrated using multiplication, it gives the performance by preserving the common pixels form all saliency maps as we seen in the previous section as shown in Figure 4.32 and Precision-Recall curve in Figure 4.30(b). The combination of colour and colour change feature also given high performance as using the information of the known target and temporal differecing is enough to detect objects.

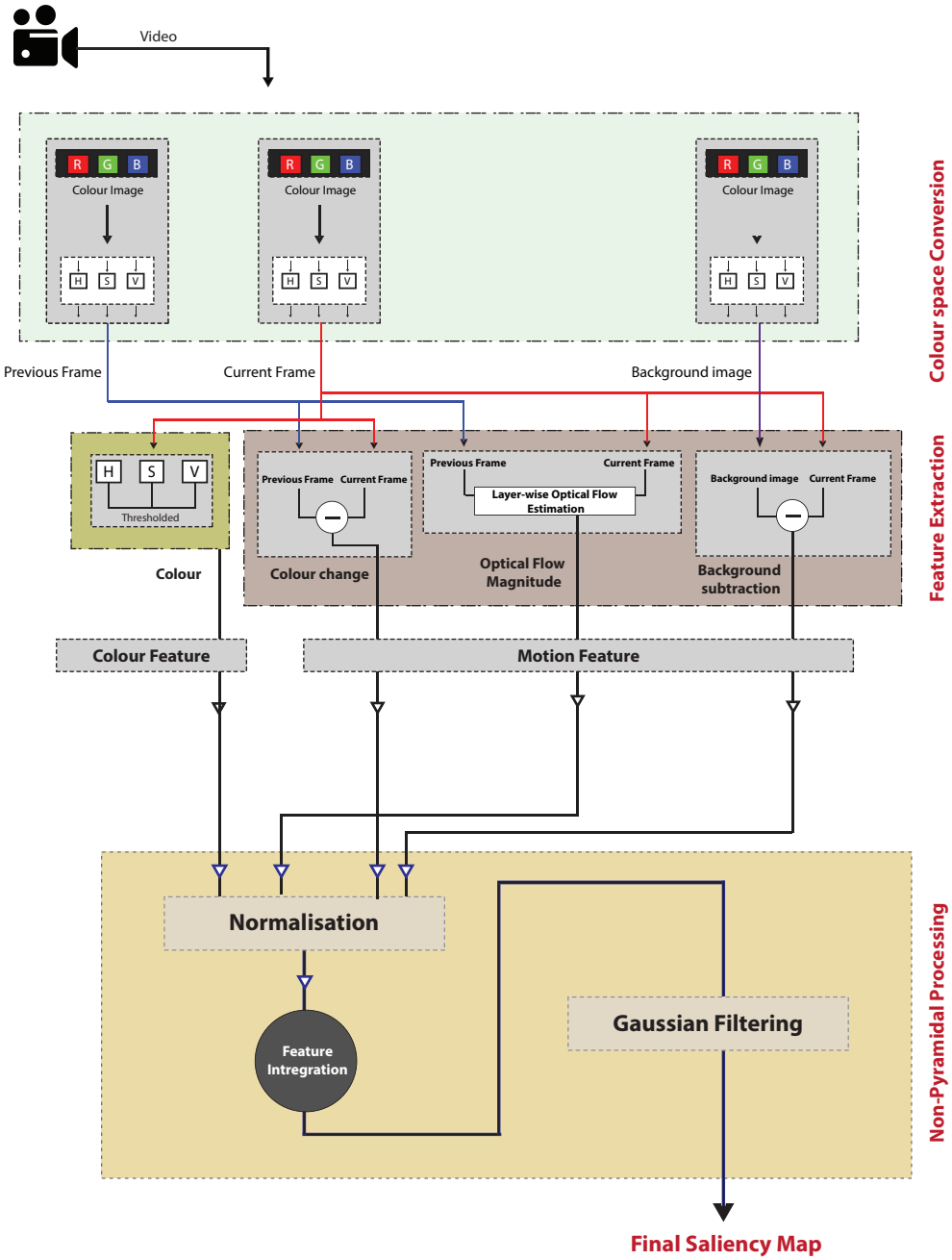


Figure 4.31: The proposed MSD model using the non-pyramidal approach.

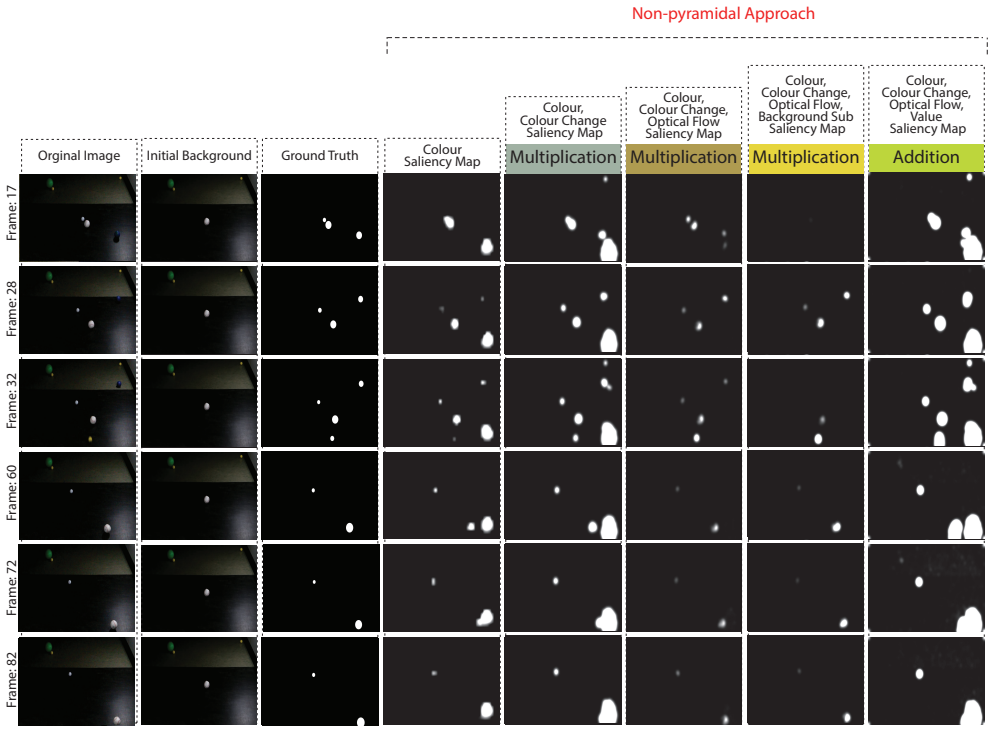


Figure 4.32: Result images of feature integration using the non-pyramidal approach.

4.6 Conclusion

In this chapter, a motion saliency detection model is developed by using different features such as colour, colour change, optical flow and background subtraction. The overall contribution of this work was to develop a motion saliency method based on visual features and temporal analysis of motion that can be incooperated with the tracking component of the system. The contributions from this research work are as follows:

1. Different challenging scenarios are created in a controlled environment with variability to the objects that are required to be detected. Different coloured objects are included with variation in the size. We constructed the ground truth of these datasets to compare the detec-

tion performance of the saliency map results. To make these scenes close to the real-world situations, we have included complicating factors such as changes in illumination, object occlusion and moving camera. We have carefully drawn the ground truth of these datasets to evaluate the performance of detection.

2. We conducted some preliminary experiments to compare the HSV and RGB colour spaces when studying the main colour and motion feature to identify which one is best to use with the proposed detection model. We have found that a HSV-based pyramidal approach is much more effective than the RGB-based pyramidal approach. This is due to the fact the that HSV colour space better replicates human perception. The detailed analysis on using the HSV-based colour space is given in Section 4.2.3.

3. Different features such as colour, colour change, optical flow and background subtraction are used to detect motion as discussed in this chapter. These features can be used as individually to detect motion or can be combined to increase the detection performance. We have presented the detailed analysis of these features in this chapter and explained how they contribute to the increase of the performance of the motion saliency system.

4. For the analysis of each feature, we have computed two approaches such as pyramidal and non-pyramidal approach. The pyramidal approach detects local contrast that works better than non-pyramidal for some of the features.

5. Feature integration is performed using the two arithmetic operations, addition and multiplication. We have found that pixel-wise multiplication is a better operation to integrate the features. The colour feature when used as an individual feature gives poor performance while generating saliency maps as it detects extra ahue in-

formation as shown in Figure 4.32. Whereas the noisy regions are eliminated when combined with the other features. It can be seen that the detection performance gets better as more features are added as shown in Precision-Recall curve in Figure 4.30(b). Overall, feature integration using a pyramidal approach is better than the non-pyramidal as it uses local contrast to detect objects. Several cross-scale combinations also help in the elimination of unwanted noise from the saliency maps.

Different experiments as discussed in this chapter shows that the proposed motion saliency model perform object detection effectively. The saliency map quality depends on the individual response of each feature. If any map is noisy then the final response can be corrupted. Some technique is required to define the accuracy in detecting the pixels from the feature maps to produce the final saliency map.

Chapter 5

Conclusions and Future Work

In this research work a saliency based multiple object tracking system was developed by combining a visual attention computational model and a Kalman filter. The overall contribution for this research was to create an effective and efficient multiple object tracking system that can replicate aspects of the human vision such as shifting attention from one target to another in a complex environment. This chapter includes detail of the contributions of this thesis and outlines future work.

5.1 Contributions

The major contributions of this thesis are as follows:

5.1.1 Active Attention Based Object Tracking System

In Chapter 3, an object tracking system is proposed that integrate a visual attention computational model and a Kalman Filter. We have evaluated the effectiveness of this proposed system by creating some simulated datasets with different challenging scenarios. This testing ensures this system can work in complex real-time situations. The details of these datasets are discussed in Section 3.3 (on page 51). We have used constant a veloc-

ity motion model in some of these scenarios where one or multiple objects moving horizontally. A constant acceleration motion model was used for projectile motion.

A multiple object tracking system was developed using a visual attention computation model based on Itti et al. [14] and integrated with the Kalman filter to track multiple objects. The proposed method was able to estimate the locations of the targets. The results showed that the estimated response approaches the true values. The Kalman gain also approached steady-state as the filter gained confidence in its estimates as explained in Section 3.2.3 (on page 56).

The object detection was driven by saliency that is used by the Kalman filter to identify the uncertainty in the state of the moving object. This uncertainty was measured by finding the area of the ellipse that was drawn by the state error covariance. This uncertainty was used to drive shifts of attention from one object to another. The simulations showed that tracker could shift attention from one object to another depending on the process noise. For instance, the mean square error between the estimated location and true value became 2.4 pixels when the same process noise was used for multiple objects. When different process noise is used for each object then the tracker shifted attention to the object with the high process noise more frequently, and the state estimates became more uncertain due to the high process noise. The mean square error when different process noise was used for multiple objects became 4.3 pixels.

Occlusion handling was tested using a scenario with multiple occluded objects. The tracker predicted the location of the occluded object, which increased its uncertainty. In the meanwhile, the attention of the tracker was shifted on another object and correct its location estimate. The results showed that the mean square error between the estimated and true locations for the occluded object was high because the tracker was not paying attention (refer to Section 3.4.2 on page 56 for details).

5.1.2 A Comparison of RGB and HSV Colour Spaces

In Chapter 4, we have performed a detailed analysis by defining the overall image content in term of the RGB or the HSV colour spaces to be used with the proposed system.

Here we have created some new datasets to demonstrate real-world environments including challenging situations such as illumination changes and camera moving, as discussed in chapter 4. In these datasets, we have used a different number of objects having various colours. These were used to measure the performance of object detection using HSV colour space instead of RGB colour space. The effectiveness of these experiments was evaluated using the Precision-Recall curve, which was used to compare the quality of the generated saliency map with the ground truth map.

The analysis compared the HSV and RGB colour spaces and identified that the HSV-based pyramidal approach was more effective than the RGB-based pyramidal approach. The main reason is that the HSV colour space replicates human perception as hue does not change with illumination and the three HSV components (hue, saturation and value) are highly uncorrelated. We have also found that by combining the Saturation-Value or Hue-Value channels, then this improves the detection of salient objects and gives a higher Precision-Recall curve. The RGB colour space gives a low precision score as it detects the object using the colour information present in an image.

5.1.3 Motion saliency model

In Chapter 4, we have developed a motion saliency model to detect salient moving objects from video. Different features such as colour, colour change between different frames, optical flow estimation and background subtraction were explored, and the appropriate feature combinations were chosen for further experimentation. We have also used the basic feature integration using arithmetic operation such as addition and multiplication

and found that multiplication is a better operation to integrate the features. The experiments revealed that the colour features gives low performance when used individually. When this colour feature integrated with other features, then performance gets betters as only the moving objects or regions get detected.

Here we have also compared two approaches such as pyramidal and non-pyramidal approach to analyse these features. When using the pyramidal integration using multiplication operation yield high performance when using all the subfeatures with background, subtraction eliminates most of the ghost region detection. The non-pyramidal integration of some of the features such as colour and colour change feature using multiplication operation showed good performance as object detection depends on the information of the known target and temporal difference.

Here we can conclude that the pyramidal approach detects local contrast that worked better than non-pyramidal for some of the features. The feature integration using a pyramidal approach was better than the non-pyramidal as it used local contrast to detect objects.

5.2 Future Work

Experiments with different scenarios, as discussed in the previous chapters show that the proposed tracking system effectively shifts attention from one object to another. In the future, the proposed model tracking system can be extended by including different features, e.g., a colour feature that can detect the different coloured objects and other features that can quickly identify objects and accurately in a complex environment.

We have integrated different feature using basic arithmetic operations such as addition and multiplication in the current system. We have observed that multiplication is a better operation to combine different features. In future, we can use dynamic feature integration that can change the number and types of features that are combined according to the sce-

nario. Some feature weighting technique will be included in the motion saliency model to increase the final saliency map quality by selecting the best feature maps.

We would like to work on the attention module of the tracking system that decides when to distribute attention among multiple objects in the scenario. There should be some method that let it decide when to search for new targets and when to pay attention to the objects already present in the scene. In the current system, we have a sequential attention shift method that is to shift attention to one of the objects in a frame. However, if the system is very sure about the objects already present in the scene, then the system could look for new objects. Here, the saliency information can be used to calculate the certainty about any object or looking for a new object.

We would also like to develop a target formation module in the system. Target formation comprises detecting and acquiring a particular target. It can include classifying the specific target according to the specific class, which useful in providing an appropriate motion model for the tracker. Here, any target can be acquired by manually pointing to the particular object or by using auto-detection methods.

Bibliography

[1] A. Kimura, R. Yonetani, and T. Hirayama, "Computational Models of Human Visual Attention and Their Implementations: A Survey." *The Institute of Electronics,Information and Communication Engineers Transactions*, no. 3, pp. 562–578, 2013.

[2] A. Borji, M. Cheng, H. Jiang, and J. Li, "Salient Object Detection: A Survey," *Computing Research Repository*, 2014.

[3] G. W. Maus, J. Ward, R. Nijhawan, and D. Whitney, "The Perceived Position of Moving Objects: Transcranial Magnetic Stimulation of Area MT+ Reduces the Flash-Lag Effect," *Cerebral Cortex*, vol. 23, pp. 241–247, 2013.

[4] W. Luo, X. Zhao, and T.-K. Kim, "Multiple Object Tracking: A Review." *Computing Research Repository*, pp. 1–36, 2014.

[5] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," *Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[6] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*, ser. Lecture Notes in Computer Science. Springer, 2006, vol. 3899.

[7] B. Y. Lee, L. H. Liew, and W. S. Cheah, "Occlusion handling in videos object tracking: A survey," ser. Institute of Physics Conference Series: Earth and Environmental Science, 2014, pp. 1–6.

[8] R. Panahi, I. Gholampour, and M. Jamzad, "Real time occlusion handling using Kalman Filter and mean-shift," *Machine Vision and Image*, pp. 320–323, 2013.

[9] S. Xu, H. Huo, and F. Tao, "Object Tracking based on time-varying saliency," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2008.

[10] E. Cuevas, D. Zaldivar, and R. Rojas, "Kalman Filter for Vision Tracking," Tech. Rep., 2005.

[11] S. Frintrop and M. Kessel, "Most salient region tracking." in *Proceedings of International Conference on Robotics and Automation* . IEEE, 2009, pp. 1869–1874.

[12] V. Mahadevan and N. Vasconcelos, "Saliency-based discriminant tracking." in *Proceedings of Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1007–1013.

[13] S. Zhang and F. Stentiford, "A saliency based object tracking method." in *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, E. Izquierdo, Ed. IEEE, 2008, pp. 512–517.

[14] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions onPattern Analysis and Machine Intelligence,*, vol. 20, no. 11, pp. 1254–1259, 1998.

[15] H. Liu and Y. Shi, "Robust visual tracking based on selective attention shift," in *Control Applications & Intelligent Control*. IEEE, 2009, pp. 1176–1179.

[16] V. Papadourakis and A. Argyros, "Multiple Objects Tracking in the Presence of Long-term Occlusions," *Proceedings of the Computer Vision and Image Understanding*, vol. 114, no. 7, pp. 835–846, 2010.

[17] Y. Huang, T. S. Huang, and H. Niemann, "Segmentation-based object tracking using image warping and Kalman filtering." in *Proceedings of the International Conference on Image Processing*, 2002, pp. 601–604.

[18] C. Wang, M. de La Gorce, and N. Paragios, "Segmentation, ordering and multi-object tracking using graphical models." in *Proceedings of the International Conference on Computer Vision*. IEEE, 2009, pp. 747–754.

[19] W. Brendel and S. Todorovic, "Video object segmentation by tracking regions." in *Proceedings of International Conference on Computer Vision*. IEEE, 2009, pp. 833–840.

[20] A. Holzbach and G. Cheng, *A fast and scalable system for visual attention, object based attention and object recognition for humanoid robots.* IEEE, 2014.

[21] R. J. Campbell and P. J. Flynn, "A Survey Of Free-Form Object Representation and Recognition Techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 166–210, 2001.

[22] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Transactions on Circuits and Systems for Video Technology,*, vol. 13, no. 11, pp. 1020–1037, 2003.

[23] S. X. Yu and D. A. Lisin, "Image Compression Based on Visual Saliency at Individual Scales." in *Proceedings of Advances in Visual Computing. International Symposium on Visual Computing*, ser. Lecture Notes in Computer Science, vol. 5875. Springer, 2009, pp. 157–166.

[24] N. Ouerhani, J. Bracamonte, H. Hugli, M. Ansorge, and F. Pellandini, *Adaptive color image compression based on visual attention*, 2001.

[25] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery." in *Proceedings of Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 853–860.

[26] T. Liu, J. S. 0001, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to Detect A Salient Object." 2007.

[27] S. Frintrop, A. Nüchter, and H. Surmann, "Visual Attention for Object Recognition in Spatial 3D Data." in *Proceedings of Workshop on Attention and Performance in Computational Vision.*, ser. Lecture Notes in Computer Science, vol. 3368, 2004, pp. 168–182.

[28] S. Li and M. C. Lee, "Fast Visual Tracking using Motion Saliency in Video." in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2007, pp. 1073–1076.

[29] Z. Ying, G. Li, S. Wen, and G. Tan, "Offset correction in RGB color space for illumination robust-image processing," *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1557–1561, 2017.

[30] R. V. Babu, P. Pérez, and P. Bouthemy, "Robust tracking with motion estimation and kernel-based color modelling." in *Proceedings of International Conference on Image Processing*. IEEE, 2005, pp. 717–720.

[31] S. Sarkka, "Bayesian Filtering and Smoothing," 2013.

[32] R. Kleinbauer, "Kalman Filter Implementation with Matlab," pp. 1–37, 2004.

[33] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Tech. Rep. 95-041, 1995.

[34] T. J. Broida and R. Chellappa, "Estimation of Object Motion Parameters from Noisy Images." *Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 90–99, 1986.

[35] D. Beymer and K. Konolige, "Real-Time Tracking of Multiple People Using Continuous Detection," in *Proceedings of International Conference on Computer Vision Frame-rate Workshop*. DARPA, 1999.

[36] K. Mantripragada, F. C. Trigo, F. P. R. Martins, and A. T. de Fleury, "Vehicle Tracking using Feature Matching and Kalman filtering," *ABCM Symposium Series in Mechatronics*, vol. 6, pp. 497–507, 2014.

[37] V. Takala and M. Pietikäinen, "Multi-Object Tracking Using Color, Texture and Motion." in *Proceedings of the Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–7.

[38] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 5, pp. 802–817, 2006.

[39] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey." *Transactions on Applied Perception*, vol. 7, no. 1, 2010.

[40] Y. Xue, X. Guo, and X. Cao, "Motion saliency detection using low-rank and sparse decomposition." in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. . IEEE, 2012, pp. 1485–1488.

[41] A. Borji, D. N. Sihite, and L. Itti, "Salient Object Detection: A Benchmark." in *Proceedings of European Conference on Computer Vision*, vol. 7573, 2012, pp. 414–429.

[42] S. Nataraju, V. Balasubramanian, and S. Panchanathan, *Learning attention based saliency in videos from human eye movements*, ser. Workshop on Motion and Video Computing, 2009, pp. 1–6.

[43] C. Siagian and L. Itti, "Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention," *Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2006.

[44] J. M. Wolfe, K. R. Cave, and S. L. Franzel, "Guided search: An alternative to the feature integration model for visual search." *Journal of Experimental Psychology: Human Perception & Performance*, pp. 419–433, 1989.

[45] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*, ser. Lecture Notes in Computer Science. Springer, 2006, vol. 3899.

[46] S. Frintrop, G. Backer, and E. Rome, "Goal-Directed Search with a Top-Down Modulated Computational Attention System." in *DAGM-Symposium*, ser. Lecture Notes in Computer Science, vol. 3663. Springer, 2005, pp. 117–124.

[47] C. Koch and S. Ullman, "Shifts in selective attention: Towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, pp. 219–227, 1985.

[48] O. L. Meur, P. L. Callet, and D. Barba, "Predicting visual fixations on video based on low-level visual features," *Vision Research*, vol. 47, no. 19, pp. 2483–2498, 2007.

[49] P. Bian and L. Zhang, "Biological Plausibility of Spectral Domain Approach for Spatiotemporal Visual Saliency." in *Proceedings of 26th International Conference on Neural Information Processing*, vol. 5506, 2008, pp. 251–258.

[50] G. Kootstra, B. de Boer, and L. Schomaker, "Predicting Eye Fixations on Complex Visual Stimuli Using Local Symmetry." *Cognitive Computation*, vol. 3, no. 1, pp. 223–240, 2011.

[51] D. Sidibé, D. Fofi, and F. Mériaudeau, "Using visual saliency for object tracking with particle filters." in *Proceedings of the European Signal Processing Conference*. IEEE, 2010, pp. 1776–1780.

[52] X. Wang, C. Liu, G. Liu, L. Liu, and S. Gong, "Pedestrian Recognition Based on Saliency Detection and Kalman Filter Algorithm in Aerial Video." in *Proceedings of the International Conference on Computational Intelligence and Security*. IEEE, 2011, pp. 1188–1192.

[53] C. C. Loy, T. Xiang, and S. Gong, "Salient motion detection in crowded scenes." in *Proceedings of the International Symposium on Communications, Control and Signal Processing*. IEEE, 2012, pp. 1–4.

[54] Y. Tong, F. A. Cheikh, F. F. E. Guraya, H. Konik, and A. Trémeau, "A Spatiotemporal Saliency Model for Video Surveillance." *Cognitive Computation*, vol. 3, no. 1, pp. 241–263, 2011.

[55] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Comparative study of background subtraction algorithms." *Journal of Electronic Imaging*, vol. 19, no. 3, 2010.

[56] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scenes." in *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–6.

[57] C. Liu, "Beyond Pixels: Exploring New Representations and Applications for Motion Analysis," Ph.D. dissertation, 2009.

[58] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*. Springer Berlin Heidelberg, 2004, pp. 25–36.

[59] L. Itti, N. Dhavale, and F. Pighin, "Realistic Avatar Eye and Head Animation Using a Neurobiological Model of Visual Attention," in

Proceedings of the International Symposium on Optical Science and Technology. SPIE Press, 2003, pp. 64–78.

[60] V. Kravtchenko, "Tracking Color Objects in Real Time," Ph.D. dissertation, 1999.

[61] M. Yokoyama and T. Poggio, *A contour-based moving object detection and tracking*. IEEE, 2005.

[62] M. Murshed, M. Kabir, and O. Chae, "Moving object tracking - an edge segment based approach," *International Journal of Innovative Computing, Information and Control*, vol. 7, pp. 3963–3979, 2011.

[63] R. Haralick, K. Shanmugam, and I. Dinstein, "Texture Features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, 1973.

[64] O. Tuzel, F. Porikli, and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification," in *Proceedings of the 9th European Conference on Computer Vision*. Springer-Verlag, 2006, pp. 589–600.

[65] I. Austvoll and B. Kwolek, "Region Covariance Matrix-Based Object Tracking with Occlusions Handling." in *Proceedings of Computer Vision and Graphics.*, vol. 6374, 2010, pp. 201–208.

[66] S. Kanprachar and S. Tangkawanit, "Performance of RGB and HSV Color Systems in Object Detection Applications under Different Illumination Intensities." in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, ser. Lecture Notes in Engineering and Computer Science. Newswood Limited, 2007, pp. 1943–1948.

[67] D. J. Bora, A. K. Gupta, and F. A. Khan, "Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Im-

age Segmentation." *The Computing Research Repository*, pp. 192–203, 2015.

[68] P. R. K. K. Shailaja Surkutlawar, "Shadow Suppression using RGB and HSV Color Space in Moving Object Detection," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 1, 2013.

[69] M. Kampel, H. Wildenauer, P. Blauensteiner, and A. Hanbury, "Improved Motion Segmentation Based on Shadow Detection." *Electronic Letters on Computer Vision and Image Analysis*, vol. 73, pp. 519–534, 2009.

[70] G. Paschos, "Perceptually uniform color spaces for color texture analysis: an empirical evaluation," *IEEE Transactions on Image Processing*, vol. 10, pp. 932–937, 2001.

[71] V. Tsagaris and V. Anastassopoulos, "Multispectral image fusion for improved rgb representation based on perceptual attributes," *International Journal of Remote Sensing*, pp. 3241–3254, 2005.

[72] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval." in *International Conference on Image Processing*. IEEE, 2002, pp. 589–592.

[73] I. Tastl and G. R. Raidl, "Transforming an analytically defined color space to match psychophysically gained color distances." in *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts*, ser. Proceedings of international society for optics and photonics. SPIE, 1998, pp. 98–106.

[74] R. Achanta, F. J. Estrada, P. Wils, and S. Süsstrunk, "Salient Region Detection and Segmentation." in *Computer Vision Systems*, vol. 5008, 2008, pp. 66–75.

[75] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency." in *Proceedings of Neural Information Processing Systems*. MIT Press, 2006, pp. 545–552.

[76] B. Yang, X. Zhang, J. Liu, L. Chen, and Z. Gao, "Principal components analysis-based visual saliency detection." in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2016, pp. 1936–1940.

[77] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global Contrast Based Salient Region Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[78] K. Deb and A. H. Suny, "Shadow Detection and Removal Based on YCbCr Color Space." *Smart CR*, vol. 4, pp. 23–33, 2014.

[79] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient Region Detection via High-Dimensional Color Transform." in *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 883–890.

[80] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search." in *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 3183–3192.

[81] C. Huang, Q. Liu, and S. Yu, "Regions of interest extraction from color image based on visual saliency." *The Journal of Supercomputing*, vol. 58, no. 1, pp. 20–33, 2011.

[82] Z. Liu, W. Chen, Y. Zou, and C. Hu, "Regions of interest extraction based on HSV color space." in *Proceedings of the International Conference on Industrial Informatics*. IEEE, 2012, pp. 481–485.

[83] S. H. Shaikh, K. Saeed, and N. Chaki, *Moving Object Detection Approaches, Challenges and Object Tracking*. Springer International Publishing, 2014, pp. 5–14.

[84] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving Target Classification and Tracking from Real-time Video," in *Proceedings of the Workshop on Applications of Computer Vision*. IEEE, 1998, pp. 8–14.

[85] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis." *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[86] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa, "A System for Video Surveillance and Monitoring," Robotics Institute, Tech. Rep., 2000.

[87] B. K. Horn and B. G. Schunck, "Determining Optical Flow," Tech. Rep., 1980.

[88] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[89] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proceedings of the Defense Advanced Research Projects Agency Image Understanding Workshop*. Morgan Kaufmann Publishers Inc, 1981, pp. 121–130.

[90] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss, "Human-assisted motion annotation," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[91] R. Andrews and B. Lovell, "Color Optical Flow," in *Workshop on Digital Image Computing*, vol. 1, 2003, pp. 135–139.

[92] J. Heikkilä and O. Silvén, "A real-time system for monitoring of cyclists and pedestrians." *Image and Vision Computing*, vol. 22, no. 7, pp. 563–570, 2004.

[93] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings of National Institute of Science, India*, vol. 2, no. 1, 1936, pp. 49–55.

[94] N. S. M. Raja, V. Rajinikanth, and K. Latha, "Otsu Based Optimal Multilevel Image Thresholding Using Firefly Algorithm," *Modelling and Simulation in Engineering*, pp. 1–17, 2014.

[95] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, 1955.

[96] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the International Conference on Machine Learning*. University of Wisconsin-Madison, 2006, pp. 233–240.

[97] S. Talebzadeh Shahrbabaki, "Contribution of colour in guiding visual attention and in a computational model of visual saliency," Ph.D. dissertation, 2015.

[98] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection." in *Proceedings of Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1597–1604.

[99] R. J. Radke, *Computer Vision for Visual Effects*. Cambridge University Press, 2012.

[100] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss, "Human-Assisted Motion Annotation," 2008.

[101] D. Sun, S. Roth, and M. J. Black, "Secrets of Optical Flow Estimation and Their Principles," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2432–2439.