UNIVERSITY OF NOTTINGHAM

Visual Tracking Over Multiple Temporal Scales

by

Muhammad Haris Khan

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Science
School of Computer Science

October 2015

Declaration of Authorship

I, Muhammad Haris Khan, declare that this thesis titled, 'Visual Tracking Over Multiple Temporal Scales' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Prediction is very difficult, especially about the future"

Niels Bohr

UNIVERSITY OF NOTTINGHAM

Abstract

Faculty of Science
School of Computer Science

Doctor of Philosophy

by Muhammad Haris Khan

Visual tracking is the task of repeatedly inferring the state (position, motion, etc) of the desired target in an image sequence. It is an important scientific problem as humans can visually track targets in a broad range of settings. However, visual tracking algorithms struggle to robustly follow a target in unconstrained scenarios. Among the many challenges faced by visual trackers, two important ones are occlusions and abrupt motion variations. Occlusions take place when (an)other object(s) obscures the camera's view of the tracked target. A target may exhibit abrupt variations in apparent motion due to its own unexpected movement, camera movement, and low frame rate image acquisition. Each of these issues can cause a tracker to lose its target.

This thesis introduces the idea of learning and propagation of tracking information over multiple temporal scales to overcome occlusions and abrupt motion variations. A temporal scale is a specific sequence of moments in time e.g. $[t-1;t]$ in an image sequence. Models (describing appearance and/or motion of the target) can be learned from the target tracking history over multiple temporal scales and applied over multiple temporal scales in the future. With the rise of multiple motion model tracking frameworks, there is a need for a broad range of search methods and ways of selecting between the available motion models.

The potential benefits of learning over multiple temporal scales are first assessed by studying both motion and appearance variations in the ground-truth data associated with several image sequences. A visual tracker operating over multiple temporal scales is then proposed that is capable of handling occlusions and abrupt motion variations. Experiments are performed to compare the performance of the tracker with competing methods, and to analyze the impact on performance of various elements of the proposed approach. Results reveal a simple, yet general framework for dealing with occlusions and abrupt motion variations. In refining the proposed framework, a search method is generalized for multiple competing hypotheses in visual tracking, and a new motion model selection criterion is proposed.

Acknowledgements

I would like to convey my sincerest gratitude to my supervisors Prof. Tony Pridmore and Dr. Michel Valstar for all the guidance and support they have provided throughout the development of this thesis with patience and knowledge. They were always available to listen my ideas, offer beneficial discussions, and provide useful feedback. They were always there to boost my confidence at the time of setbacks. Without their advice and unwavering support, this dissertation would not have been possible.

I am grateful to all my colleagues at Computer Vision Laboratory at Nottingham University for providing a friendly and a nice working atmosphere.

Last but not least, I would like to thank my beloved parents and my dear wife for their love and support, which has given me the confidence to overcome difficulties through the years. You were simply phenomenal in understanding and supporting me for whatever I aspired to achieve in my life.

Contents

List of Figures

List of Tables

To my beloved parents and dear wife...

Chapter 1

Introduction

Computer vision has been studied actively as a scientific field since the late 1960s, when computing power started to increase rapidly. Computer vision started with the goal of building a system that can process images in a similar way to the human visual system, however, it has evolved into a much broader field. For instance, applications like image database search in the world wide web, biometrics, and computational photography have arisen from developments in computer vision over the years. In addition, the proliferation of digital video cameras in daily life is generating large amounts of visual data. In response to large volumes of data, there is a growing demand for robust visual analytics algorithms to perform automatic analysis either in real-time or offline.

To be able to understand video content fully, it is important to know certain information about the target(s)/object(s) present in it. This information could be where each target is, what it is doing etc. The answer to the two questions mentioned lies in how it is moving i.e. in establishing its correspondence from one frame to the next, which is known as visual tracking, object tracking, or target tracking. Visual object tracking is a fundamental component in many computerized video applications such as intelligent traffic control [Hsieh et al., 2006, Morris and Trivedi, 2008, Zhou et al., 2007], security and surveillance [Chen et al., 2011, Hampapur et al., 2005] human-computer interaction [Poole and Ball, 2006, Wang et al., 2006], event detection and recognition [Johnson and Hogg, 1996, Saleemi et al., 2009, Smith et al., 2006], and many others. Due to its large potential impact, visual tracking has remained an active topic in computer vision for more than two decades.

Specifically, visual tracking has been researched actively since the mid 1980s, although many estimation tools and statistical methods used to solve this problem were developed long before that time [Smith, 2007].

Out of the many routes to visual tracking, perhaps the most popular is the probabilistic approach. At a very basic level, a probabilistic approach makes use of the information available from the video, and prior knowledge about the target (object) to build a probability distribution, which can be conceived as the belief about the target's state. The target state is typically a set of parameters such as position, orientation and scale of a bounding box used to represent the target. This probability distribution is updated as soon as information, in the form of a new video frame, arrives.

Probabilistic approaches have remained popular in the realm of visual tracking because of their ability to handle uncertainties in the measurements and models involved in a principled manner. While tracking under uncontrolled conditions, uncertainties may result from varying target appearance caused by factors such as illumination changes in the scene, and variations in apparent target motion caused by e.g. low frame-rate image acquisition.

The work presented in this thesis focuses on a widely celebrated probabilistic approach to tracking known as recursive Bayesian estimation (RBE). It maintains a (unknown) probability distribution, which encodes knowledge about the target state given information from the video. This probability distribution is estimated in a recursive fashion over time through extracting information from the incoming video frames.

Although many difficulties in visual tracking can be mitigated by a multi-camera setup, the work in this thesis is restricted to visual tracking through a single camera system, as most existing systems are equipped with a single camera.

1.1 Motivation for Study

Despite the fact that visual tracking is a well-defined computer vision problem, it is challenging. It remains unsolved due to the inevitable variations associated with the tracked target, and occlusions.

Appearance Variations: Tracking algorithms require some kind of model to detect the presence of a target in each frame of an image sequence. The model describes the appearance of the target i.e. what it looks like. Usually, it is built prior to the start of tracking by utilizing information from the first frame. As it relies on limited information, this appearance description might not be adequate when the target's appearance changes over time. Such change can be the result of variation in scene lighting, target pose, and target shape (due to non-rigid deformation). Under these circumstances, the model needs to adapt to these variations over time to remain an appropriate description of the target's true appearance.

On the other hand, this adaptation risks the tracker losing its target due to the accumulation of imperfect tracking results over time, which leads to a gradual mis-representation of the target's appearance. Although many methods have been presented to address this adaptivity versus stability problem, it is still an open problem. Most of the solutions presented so far are conservative in approach as they do not allow abrupt appearance variations to be captured.

Complex Motion Variations: In unconstrained environments, a target can display complex motion variations, which can be smooth or abrupt. To achieve efficient tracking, it is typically assumed that the motion of the target varies smoothly over time. However, under realistic tracking scenarios, a target's motion can change abruptly over a given time interval due to factors such as camera motion/switching, low frame rate image acquisition, etc.

A vital component of probabilistic trackers is a motion model, which describes the movement of a target over time. Assuming smooth motion variations, most of the existing tracking approaches employ motion models that exploit a single temporal scale, e.g. Random-Walk (RW). These models use the most recent state to predict the state at the current time. A temporal scale is a specific sequence of moments in time e.g. $[t-1]$, $[t-1;t]$, or $[t-3;t]$. When a target exhibits a range of motion patterns(smooth to abrupt), single scale models may find it difficult to handle this multi-modal variability in motion. Building motion models by exploiting multiple recent state histories could, however, provide a way to cover variable target motion better than single scale models.

Tracking frameworks use search methods to seek for the most likely hypothesis from the space of all available hypotheses. Here, the hypothesis is the state of a target. Tracking algorithms have typically deployed a single motion model (like RW), and so relied upon a few search methods (like Particle Filters). To cover different motion patterns, which single motion model might not be capable of, multiple motion model methods have attracted increased interest in the recent past. The existence of multiple hypotheses produced by these motion models necessitates that a broader range of search methods be used in visual tracking.

The presence of multiple motion models also raises an important question: how to pick the most suitable hypothesis, i.e. closest to the true target state, as the hypotheses generated by all the models are usually not equally accurate. Exploration of the space of possible motion model selection criteria is a prospective research direction.

Occlusions: Occlusions arise when the view of the camera that tracks the target is obscured by another object or objects. Although it is a classic problem in visual tracking, it remains unsolved due to the difficult nature of the problem itself. This is because target

observation (image-based evidence) becomes very weak, or is fully unavailable, during occlusions. Furthermore, occlusions can take place over different periods of time.

Most of the existing probabilistic tracking algorithms predict a target's state at time t using only its state at time $t-1$. In other words, they all operate on a single temporal scale $[t-1;t]$. This is a reasonable approach to tracking at time t as long as the immediate previous state (at time $t-1$) is accurately estimated. However, in the presence of problems such as occlusion, it is quite possible that the estimated state at time $t-1$ might be completely different to the true underlying state. This inaccurate target state at time $t-1$ may make it difficult to estimate the true target state at time t since a large tracking error will be propagated. As a result, the tracker can start to or completely lose the target.

Extending the tracker's temporal scale alone might not be enough to adequately cover the variation in periods of occlusions. However, a tracker operating over multiple temporal scales may be able to solve this problem. Such a tracker will make use of several previous estimated states instead of only one at time $t-1$ to infer target state at time t.

1.2 Contributions

The work described in this thesis proposes to exploit multiple temporal scales to address two important and outstanding problems in visual tracking: occlusions and abrupt motion variations. The central objective of the thesis is to investigate the usefulness of multiple temporal scales of model generation and application to deal with the problems related to occlusions and abrupt motion variations. In addition, work in this thesis generalizes a search method for multiple competing hypotheses in visual tracking, and proposes a new way of (motion) model selection. In brief, the work presented in this thesis makes the following contributions:

- The thesis assesses potential benefits of multiple temporal scales to visual tracking using ground truth data. Specifically, both motion and appearance variations are investigated by doing preliminary, but thorough experiments on the ground truth data of several image sequences.
- The thesis proposes a visual tracker operating over multiple temporal scales that is capable of handling occlusions and abrupt motion variations. This is accomplished by learning motion models from the target history at different temporal scales, and applying those over multiple temporal scales in the future. An extension of the bootstrap particle filter is presented to search around the predictions generated by motion models.

- The proposed tracking framework is compared to state-of-the-art tracking algorithms (both according to the CVPR'13 [Wu et al., 2013] benchmark, and their capability to handle occlusions and abrupt motion variations) on both publicly available benchmarks and some new data. Both quantitative and qualitative evaluations reveal that the proposed framework shows favourable performance against competing methods. Experiments are also conducted to analyse the performance of the proposed framework by excluding vital components, keeping some parameters fixed, and varying the degrees of polynomial motion models.
- With the goal of exploring the proposed tracking framework, and as a first step towards improving its performance further, the thesis generalizes a search method for multiple competing hypotheses in visual tracking and proposes a new (motion) model selection criterion.

1.3 Thesis Organization

This dissertation proposes and evaluates the idea of visual tracking over multiple temporal scales, and then further explores the proposed tracking framework. In the four chapters that constitute the body of this thesis, the following topics are covered:

1.3.1 Preliminary Background of the research related to this thesis

Chapter 2 begins by briefly describing the main challenges associated with visual tracking. These include change in target appearance, abrupt motion variations, and occlusions (by other target objects and the environment). It then outlines the general framework for visual tracking, and recalls some of the important and popular estimation tools used, with a special focus on recursive Bayesian estimation. Next it features an overview of the state-of-the-art approaches which are in some sense connected to the work described in this thesis. It then provides an in-depth discussion of the existing work related to the problems addressed in this thesis. Finally, the chapter concludes by summarizing important trends in prior work, and highlighting the outstanding problems in visual tracking and drawbacks in the connected body of works that propelled the research described in this thesis.

1.3.2 Visual Tracking Over Multiple Temporal Scales

Chapter 3 defines the proposed approach of visual tracking over multiple temporal scales, and assesses the potential benefits of multiple temporal scales. In particular, it examines

whether, by using a set of models, learned over different temporal scales, it is possible to generate a better prediction performance than is possible when using a single model from this set. The chapter begins by emphasizing the importance of multiple scales in the spatial domain, which served as a motivation for utilizing multiple temporal scales in visual tracking. Next it investigates both motion and appearance variations in the ground truth data of several sequences to highlight the potential benefits of access to a set of models, derived from multiple temporal scales. This chapter concludes that prediction performance can be improved if a set of models, extracted from multiple temporal scales, is combined with an ability to pick the right model from the set.

After conducting a preliminary study on the potential benefits of learning over multiple temporal scales in Chapter 3, Chapter 4 proposes a visual tracker operating over multiple temporal scales that is capable of overcoming occlusions and abrupt motion variations. The proposed tracker is compared with competing methods on both publicly available benchmarks and some new data. Experiments have also been carried out to analyze the impact on the performance of the proposed tracking framework of excluding important components, varying the degrees of the polynomial motion models used, and keeping an important parameter fixed.

1.3.3 Exploration of the proposed tracking framework

Chapter 5 develops further insight into the proposed tracking framework and takes a first step towards further improving the performance of the method. In particular, it generalizes a search method to estimate the best hypothesis around multiple competing hypotheses, and proposes a new motion model selection criterion. A modified tracker based on the generalized search method is evaluated and its performance is reported. With the aim of further exploration, the chapter then defines the model selection problem in the context of the proposed framework, and proposes a new motion model selection criterion. A further tracker based on this new selection method is assessed and then discussed. Finally, the chapter concludes by describing important findings in this exploration study.

1.4 Related Publications

The research described in this thesis has been a collaborative effort with my supervisors, Tony Pridmore, and Michel Valstar, who have provided a guiding hand in all of the work presented in this thesis.

The following publications are derived from this thesis:

1. Haris Khan, Michel Valstar, and Tony Pridmore, "MTS: A Multiple Temporal Scale Tracker Handling Occlusion and Abrupt Motion Variation", Proc. Asian Conference on Computer Vision (ACCV), 2014.

2. Haris Khan, Michel Valstar, and Tony Pridmore, "A Generalized Search Method for Multiple Competing Hypotheses in Visual Tracking", Proc. International Conference on Pattern Recognition (ICPR), 2014.

Chapter 2

Related Work

This chapter reviews important computational tools used in visual tracking, features an overview of recent state-of-the-art work in visual tracking, and provides an in-depth discussion of prior work relevant to this thesis.

It begins by describing the general visual tracking problem and highlighting major issues associated with it. It then outlines the main components of a tracking system, underscores a few early and some recent state-of-the-art work in visual tracking related to this thesis, and recalls classic tools (search methods) used to infer solutions to the tracking problem with a special emphasis on the approximation methods for recursive Bayesian estimation (RBE). RBE is a popular probabilistic approach to visual tracking, and has been used throughout the work described in this thesis. Next, it provides a detailed discussion of prior work relevant to this dissertation. As there is an enormous amount of published work on visual tracking, this chapter is not aimed at providing an exhaustive literature review. Instead, the chapter highlights bodies of work that are either related to this thesis in some sense, or are considered seminal in the visual tracking field. The chapter concludes by summarizing important trends in prior work, and outlining the outstanding problems that motivated our research.

2.1 The Visual Tracking Problem

In computer vision, visual tracking refers to the task of estimating the state of a target of interest at each time instant in a given image sequence. The target state is usually comprised of parameters such as velocity, location, and scale of a bounding box used to represent the target. Although visual tracking extends to image sequences formed by fusing outputs from multiple cameras, the scope of this dissertation is limited to sequences captured from a single camera.

At this stage, it is worthwhile to distinguish the tasks of visual object detection and recognition from visual tracking. Object detection aims to find the locations and configurations of objects of a certain class in a given image [Vedaldi et al., 2009],[Everingham et al., 2010]. A common example of this is face detection [Rowley et al., 1998],[Viola and Jones, 2004]. The detection task does not consider the temporal relationships among a sequence of images. Object recognition refers to categorizing a given object in an image as belonging to one of a set of object classes [Vedaldi et al., 2009],[Yang, 2009]. For instance, identifying the make and model of a car given different possible categories [Pearce and Pears, 2011], [Chen et al., 2015].

Although a visual tracker detects and recognizes a certain object, it further establishes the object correspondence from one frame to the next frame by introducing a temporal factor. An example of this would be to consistently infer the position, pose and motion parameters of a moving car in some image sequence. A human equivalent of tracking is to follow something with the eyes over a certain time period [Smith, 2007].

2.1.1 Online and Offline Tracking Methods

Tracking methods can be broadly classified as online or offline [Smith, 2007],[Hong and Han, 2014]. Online methods only have access to the present and past member of the image sequence, whereas offline methods can use all the information available in an image sequence. Fig. 2.1 illustrates the difference between the two tracking methods [Smith, 2007]. Many tracking applications such as video surveillance and human-computer interaction require online processing. On the other hand, large amounts of archived video in local hard drives and video sharing websites can be accurately analyzed using offline methods[Hong et al., 2013].

Most of the tracking methods discussed in this and the following chapters are online. However, a few relevant offline methods are also reviewed in the subsequent sections.

2.1.2 General Framework for Visual Tracking

In simple terms, a tracking algorithm is composed of a *matching* and a *search* strategy [Yang et al., 2009b] applied at each moment in time to find the best possible target configuration (hypothesis) in a search space. For a given tracking problem, the set of all possible solutions make up its search space [Smith, 2007]. The matching strategy quantifies how well a given hypothesis matches the available image data, while the search method seeks the optimal hypothesis through maximising or minimising an objective function, which itself is a function of the matching strategy.

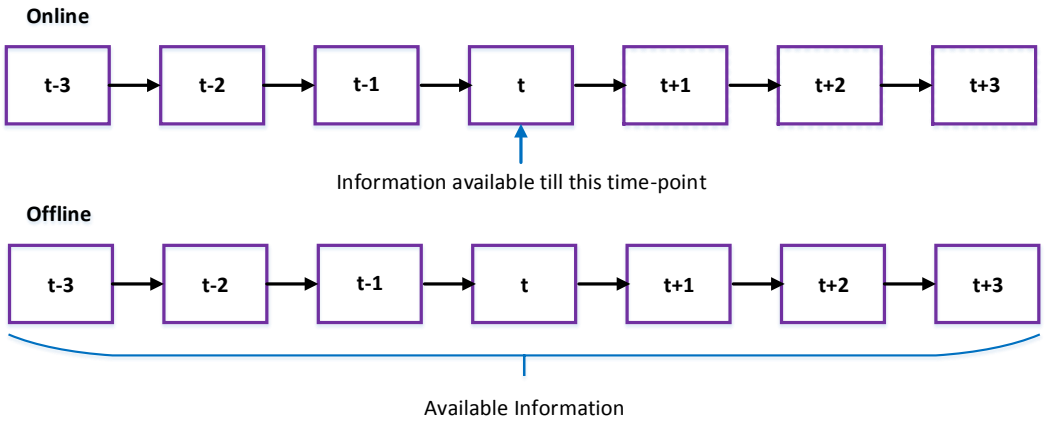


FIGURE 2.1: **Difference between online and offline tracking methods.** Lets assume that the most recent frame available for processing belongs to time t. In online methods, the information up to and including time t is available for processing, and information after time t is not. Offline methods have no restriction in terms of availability, and they can choose to track any frame from the whole image sequence regardless of temporal order.

The matching strategy, most commonly known as the observation model, is constructed based on an appearance model. Appearance modelling defines possible ways of describing a target to be tracked using information from an image sequence. This information is typically features extracted from an image sequence such as colour, texture, edges, and motion [Perez et al., 2004]. Section 2.2 introduces simple appearance models, one of which is used quite frequently in the proposed work, and later reviews some recent state-of-the-art work in adaptive appearance modelling.

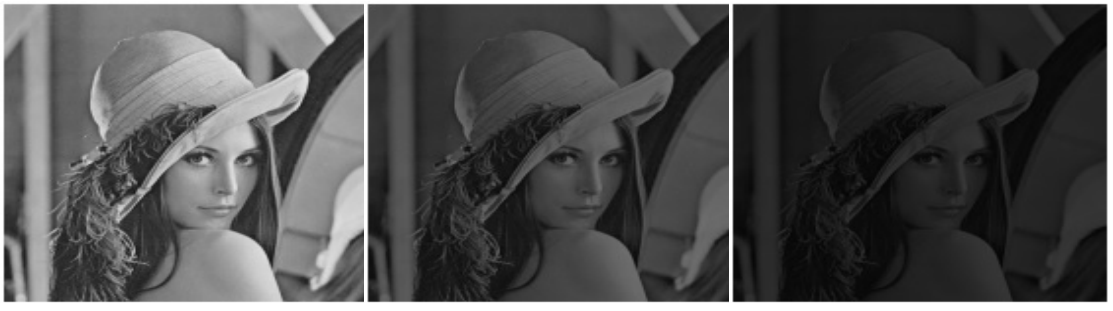
Search strategies control the way these appearance models are used in tracking to find the optimal hypothesis [Smith, 2007]. At a very primitive level, these search strategies can be classified as either probabilistic or non-probabilistic [Zhou et al., 2004]. Non-probabilistic approaches, typically known as deterministic approaches, formulate the tracking problem in terms of a cost function and use optimization techniques to minimise or maximise this function [Lucas et al., 1981],[Comaniciu et al., 2003],[Fan et al., 2010],[Sevilla-Lara and Learned-Miller, 2012]. In contrast, probabilistic approaches use statistical methods to model the uncertainty associated with the motion and appearance of the target and represent belief about the target state by fusing these information sources [Isard and Blake, 1998a],[Perez et al., 2004],[Pérez et al., 2002],[Ross et al., 2008],[Mei et al., 2011]. Section 2.3.1 reviews some of the popular non-probabilistic search methods in tracking. Section 2.3.2 discusses various probabilistic search methods in detail, as they are central to this thesis.

2.1.3 Major Visual Tracking Problems

Like object detection and object recognition, visual tracking has to deal with problems that trouble computer vision systems in general [Arvind Ganesh and Ma, 2011], such as appearance variations, background clutter and occlusions. In addition to these three challenges, a visual tracking method has to cope with variations in the apparent motion of the tracked target. In the context of visual tracking, appearance variations can be caused by changes in illumination, pose, and shape, occlusions can be partial or full, and motion can vary smoothly or abruptly in a given time interval.

Recent research in visual tracking has focussed on challenging real-world tracking problems more than the experiments in purely laboratory settings [Yilmaz et al., 2006]. In these realistic conditions, it is quite possible for a target to display any combination of the aforementioned problems. As a result, robust tracking becomes a severely complicated task in such conditions. To be able to develop a tracker that can handle real-world challenges, first it is important to understand the nature and source of these challenges in detail.

Illumination changes are one of the most important sources of appearance variations. They can influence the appearance of the target in an image as the colour of the target may change due to the properties (intensity and colour) of the falling incident light [Smith, 2007]. For instance, a target may look different under light from flashing blue and red bulbs than under sunlight. This can introduce problems since the most common models of target appearance are simple, e.g. colour distributions. As an example, Fig. 2.2 shows some images of a target captured under different illumination conditions.



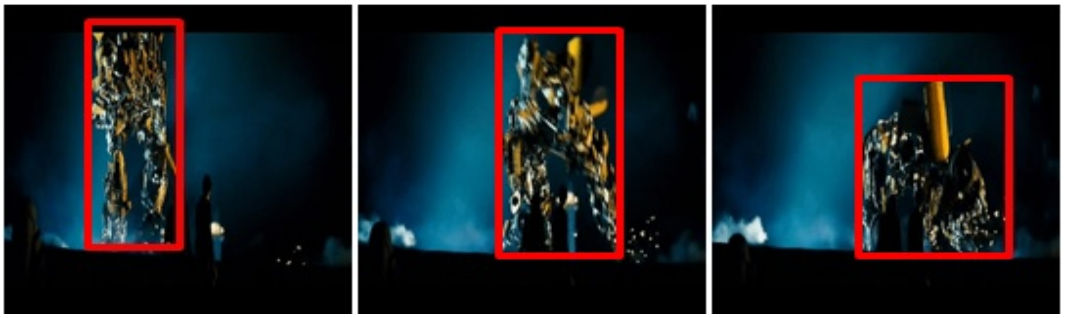
(A) (B) (C)

FIGURE 2.2: **Appearance changes due to different illumination conditions.** This figure shows three different images of a lady captured under different illuminations. These images are taken from [He et al., 2013].

Target appearance can vary significantly if it changes shape on its own. Some targets, such as cars, are rigid, while others, like humans, are non-rigid. Rigid targets do not change their shape (Fig. 2.3(a)), while non-rigid can deform and take many complex

shapes (Fig. 2.3(b)). To accurately capture the variations shown by a non-rigid target, a high dimensional state representation is required, and searching for the best possible configuration in this space can be a daunting task. For instance, the pose space in articulated human tracking has many degrees of freedom and, therefore, advanced optimization techniques are required to efficiently find the most likely configuration.

(A) Frames # 6,26, and 226 of the car sequence.



(B) Frames # 25,51, and 65 of the transformer sequence.

FIGURE 2.3: **Example of a rigid and a non-rigid target.** The target in all frames is represented by a red bounding box. (A) shows that a rigid target (car) does not change its shape during tracking, while (B) indicates that a non-rigid target (transformer) takes complex shapes during tracking.

Another dimension of appearance change is pose variation, which can be divided into in-plane and out-of-plane rotations. In-plane rotations occur when a target changes its orientation in the image plane, while out-of-plane rotations result when a target rotates across the image plane. Out-of-plane rotations can be difficult to handle since some or all of target features becomes invisible, depending on the degree of rotation. An example of appearance variation caused by out-of-plane rotations is illustrated in Fig. 2.4. For robust tracking, it is necessary to learn these variations offline or adapt to them while tracking.

Background clutter can introduce problems for trackers as it contains elements that may bear similarity to some target features. In severe cases, some patches might appear very similar to the target. These are termed distractors. The appearance model used should be *discriminative* i.e. able to confidently distinguish between the target and these

(A) (B) (C)

FIGURE 2.4: **Changes in appearance due to out-of-plane rotations.** An example of how a person's face appears differently when undergoing out-of-plane rotation. These images are taken from [Yang, 2008].

visually similar objects. Fig. 2.5 shows an example of distractors in the background while tracking a person's face.

(A) (B) (C)

FIGURE 2.5: **An example of distractors in the background while tracking a person's face.** In all three images, the target(face of a person) is represented by a yellow bounding box, while the distractors are two faces in the background.

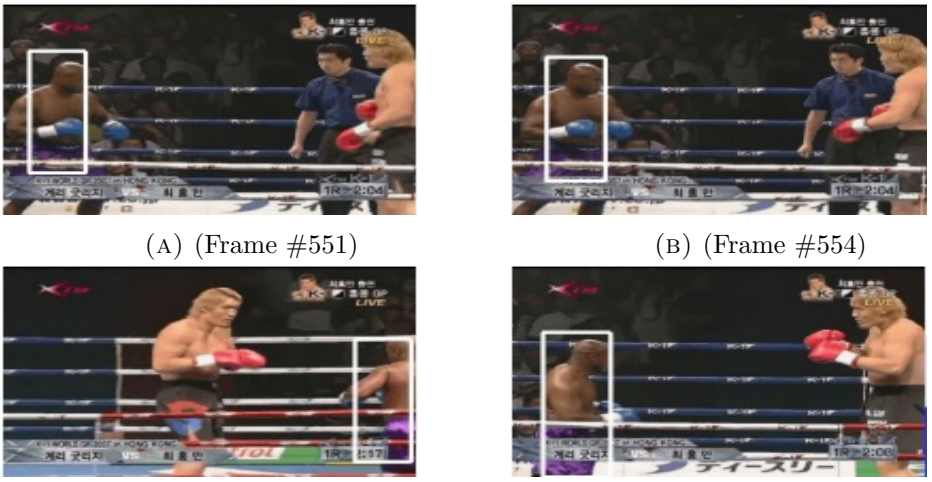
Another difficult tracking problem is occlusion. Occlusions arise when the camera's view of the tracked target is blocked by one or more stationary or moving object(s). In addition to this, targets can occlude themselves by undergoing out-of-plane rotation or articulated motion. Fig. 2.6 illustrates three different kinds of occlusion. In real-world scenarios, a tracked target can stay partially or fully occluded for variable time periods. Because the target image evidence is partially or wholly unavailable, it becomes hard for a tracker to maintain contact with the target through occlusions. This is particularly problematic if a change in the appearance or direction of motion occurs while the target is occluded. Reliable recovery of the target after occlusions is essential to achieve accurate tracking.

Although a target can exhibit numerous types of motion, in general, its motion is categorized as varying smoothly or abruptly over some time interval. Fig. 2.7 demonstrates two typical motion variations. Abrupt variations can be due to motion of the target

(A) (B) (C)

FIGURE 2.6: **Various types of occlusions.** A tracked target is represented by a cyan bounding box in (a) and (b), and by a dashed yellow bounding box in (c). The target is occluded by a stationary object in (a), and by a moving object in (b). In (c), the target is self-occluded. Image in (c) is taken from [Smith, 2007].

itself and/or camera movement. The search in state space can be constrained considerably when motion is smooth. However, to cover abrupt motions it is essential to search a relatively large area of or even the entire state space. The selected search method therefore directly determines the efficiency of the tracking method, and the robustness of the tracker to noise and local maxima.



(A) (Frame #551) (B) (Frame #554)

(C) (Frame #448) (D) (Frame #449)

FIGURE 2.7: **Two different kinds of motion variations.** This figure demonstrates smoothly and abruptly varying motion of a target, which is represented by a white bounding box in all four frames of a video sequence. From frame #551 to frame #554, the variation in motion is smooth, while from frame #448 to frame #449, it is abrupt. These images are taken from [Kwon and Lee, 2008].

Efficiency is an important requirement, particularly for online tracking methods. Because these methods are used or expected to work in real-time applications, they have to process information upon its availability as quickly as possible. On the other hand, a high-dimensional target representation coupled with online learning may be required

to achieve robust tracking. Consequently, it becomes very hard for the search mechanism to fulfill the efficiency requirement and at the same time find the best possible configuration.

2.2 Appearance Modelling

As mentioned earlier, appearance modelling characterizes possible ways of representing the target given information from a video sequence. An appearance model can be as simple as a colour histogram in some colour space, as proposed by [Pérez et al., 2002], or it can be complex, such as the hybrid generative discriminative model proposed by [Yu et al., 2008].

Before probing deeply into the literature on appearance modelling in visual tracking, it is important to briefly survey classic ways of representing target shape, as it is fundamental to appearance representation. Some typical shape representations used in visual tracking are illustrated in Fig. 2.8 and are described below.

Points are often used to represent targets that cover a small region of an image [Yilmaz et al., 2006]. Algorithms using this representation only estimate the translation of the target. This estimation can be achieved in three different ways: (1) by tracking on a frame-to-frame basis [Lucas et al., 1981], (2) through matching key-points [Veenman et al., 2001], and (3) by learning linear predictors [Zimmermann et al., 2009].

Geometric shapes are often used to describe rigid targets, and sometimes deformable targets. Common geometric shapes are the bounding box and ellipse. Methods based on this representation estimate location, scale, and in-plane rotation of the target.

Contours are often utilized to represent non-rigid targets as they delineate the target boundary. Parametric representations of contours have been applied to track human heads [Birchfield, 1998], while non-parametric representations have been used to track people [Yilmaz et al., 2004].

Articulated models are used to represent the motion of non-rigid targets which themselves are composed of rigid parts. These rigid parts are described by geometric shapes and their relative motion is governed by a model of their geometric relations [Kalal, 2011]. Articulated models have been used to track non-rigid targets such as humans [Wang et al., 2003],[Ramanan et al., 2007].

The work described in this thesis uses a bounding box to represent the target shape. This representation, though simple, has good expressive power. Typically, three parameters are used to describe the bounding box. In the subsequent sections, existing work on

(A) (B) (C) (D)

FIGURE 2.8: **Typical target shape representations used in visual tracking.** Four different shape representations shown in this figure are: (a) points, (b) geometric shapes, (c) contours, and (d) articulated models. These images are taken from [Kalal, 2011].

appearance modelling, mostly related to this shape representation, will be discussed. The discussion of appearance models has been divided into two major categories: generative models and discriminative models.

2.2.1 Generative Models

Generative models encode the appearance of the target. The simplest and perhaps the most popular generative models used in visual tracking are colour histograms. Colour features have reasonable expressive ability and are readily available from an image sequence. [Comaniciu et al., 2000] have shown that colour histograms can be used to track various targets in an efficient and robust manner. However, it is obvious that this representation cannot distinguish among targets having similar colour distributions. A remedy to this problem was proposed by [Pérez et al., 2002]. While modelling target appearance, they considered the spatial arrangement of colour features by dividing the target shape into different subregions and extracting a colour histogram from each.

Template based appearance models represent the target using a single exemplar e.g. an image patch [Schweitzer et al., 2002],[Reddy and Chatterji, 1996]. As target appearance can vary significantly over time a fixed template, formed prior to the start of tracking, cannot stay valid over extended periods of time. To adapt to these variations, it is essential to update this template during tracking. On the other hand, the adaptation is prone to *drift* as incorrect estimates of the target state can gradually corrupt the evolving template with the introduction of the background information. Moreover, a single template obviously does not allow encoding of multiple possible appearances.

To handle the trade off between adaptive and non-adaptive tracking, it is important to update the template only if necessary and re-use previously learned templates otherwise. This idea was demonstrated successfully in the work of [Matthews et al., 2004].

To represent multiple appearances of a target, a single template can be decomposed into multiple bases using a dimensionality reduction technique such as Principal Component Analysis (PCA). [Black and Jepson, 1998] utilized eigenbasis to encode known target appearances in an offline trained target model and used it to track deformable targets. However, these models are limited to tracking applications for which the target appearances are known in advance. Ross et al.[Ross et al., 2008] learned a low-dimensional subspace (eigenbasis) in an incremental manner during tracking and demonstrated robustness to moderate illumination and appearance variations. However, the appearance model can drift when presented with noisy updates, which are caused by imperfect tracking results, because the algorithm does not include any mechanism to reduce the impact of these updates.

Another way of handling variations in target appearance is to construct multiple simple appearance models, with each model responsible for covering a certain type of variation. The seminal work of Kwon and Lee [Kwon and Lee, 2010] used Sparse Principal Component Analysis (SPCA) to form multiple basic target models to track robustly in challenging tracking environments. Each target model is a mixture of templates and its cardinality is determined by the number of non-zero entries in the principal component corresponding to this target model. By sampling these basic appearance models according to the state of the recent tracking environment, [Kwon and Lee, 2011] showed improved accuracy and efficiency in real-world settings. To handle abrupt appearance variations, Park et al. [Park et al., 2012] modelled the probabilistic dependency between sequential target appearances. They clustered formerly seen target appearances based on their visual similarity, learned cluster-specific classifiers as multiple appearance models, and then modelled the dependency between these learned appearance models. The algorithm is effective in tracking an abruptly varying target appearance, but is computationally expensive.

The success of sparse representations in face recognition motivated its application in visual tracking. Mei and Ling [Mei and Ling, 2009] modelled the target as a sparse linear combination of target and trivial templates using L1-minimization. A good target candidate can be well approximated by the trivial templates leading to a sparse coefficient vector, whereas a bad candidate will often produce a dense coefficient vector. The resulting tracker shows robustness against clutter and illumination changes. However, the computational complexity of L1-minimization is high, which precludes its applicability in real-time scenarios. To improve the efficiency of sparse coding methods, [Li et al., 2011] used an orthogonal matching pursuit algorithm, and [Bao et al., 2012] proposed an accelerated proximal gradient approach to solve the costly optimization problem.

Recently, sparse representations have been further exploited, in different formulations, to enhance robustness to appearance variations. Zhang et al. [Zhang et al., 2012b] proposed a tracking framework based on multi-task sparse learning, in which the inter-dependencies among different particles were taken into account to improve the accuracy of [Mei and Ling, 2009]. In [Jia et al., 2012], a local sparse appearance model was proposed that utilizes partial as well as spatial information of the target to achieve better performance than methods based on holistic representations such as [Mei and Ling, 2009]. A dictionary, formed by sampling overlapped local image patches from a set of templates, was used to encode local patches in a candidate region. An alignment-pooling method was proposed to reduce the impact of outliers and retain structural information in this local representation.

Multi-region representations of the target are more robust to noise and other appearance variations compared to holistic, as mentioned earlier in this section. They were further used to good advantage by many approaches attempting to cover large geometric appearance changes. In [Shahed Nejhum et al., 2008], a constantly changing foreground shape was represented by a few rectangular blocks, whose positions within the tracking window were adaptively determined. To track a given frame, the algorithm finds a tracking window by scanning whole image, segments the precise boundary of the target in this window, and updates the configuration of rectangular blocks with this extracted boundary. Kwon and Lee [Kwon and Lee, 2009] proposed a more flexible patch-based appearance model, in which the patches are added, removed, and moved by affine transformation and transition to deal with large geometric and photometric appearance variations. However, the local appearance model can still drift because it does not have a mechanism to identify noisy samples during the appearance update process. Cehovin et al. [Cehovin et al., 2011] combined global appearance of the target with the local patch-based target representation using a novel coupled-layer visual model to reduce noisy samples while updating the local appearance. The aforementioned approaches have shown good accuracy while tracking non-rigid targets, but at the cost of speed.

Despite the fact that these online generative models have demonstrated success, there are two major concerns that require attention. First, a large number of examples are required from some sequence of recently processed frames to capture the likely appearance variations. Since there are often only a few available, these models assume that the target appearance does not vary significantly in this period. However, if this assumption is violated, the appearance model may again drift. Second, these models do not take into account the available background information, which again might improve their tracking accuracy and robustness [Zhang et al., 2012a].

2.2.2 Discriminative Models

Discriminative models learn a decision boundary between foreground (target) and background (non-target) examples. This decision boundary is also termed a binary classifier. The classifier can be trained offline by providing all possible target(positive) and non-target(negative) examples(patches), or can be adapted online by utilizing the weakly labeled data generated by a tracker.

For discriminative models to stay valid over extended periods of tracking, it is necessary to update the decision boundary as soon as new tracking data becomes available. Collins et al. [Collins et al., 2005] developed a framework to select the most discriminative features online and Avidan [Avidan, 2007] proposed a method to update an ensemble of weak classifiers to separate the target from the background. In [Collins et al., 2005], the set of candidate features is comprised of linear combinations of red, green, and blue pixel values. A likelihood image is produced for each candidate feature in which object pixels have positive values and background pixels have negative values. Then, a variance ratio is computed from histograms of these likelihood values to quantify the separability of object and background classes under this feature. Finally, top N most discriminative features are selected for tracking based on their ability to separate between object and background. In [Avidan, 2007], a weak classifier is a separating hyperplane learned from target and non-target examples. A strong classifier is learned using (offline)Adaboost algorithm from the ensemble of weak classifiers, which is then used to produce a confidence map in a given frame for tracking. In [Grabner and Bischof, 2006], Grabner & Bischof proposed an online version of Adaboost to select discriminative features. Again, features are weak classifiers, which perform better than a chance. However, contrary to offline boosting, where all examples are used to update one weak classifier, in online boosting, all weak classifiers get updated upon the arrival of a new example.

Although the aforementioned approaches adapt to target appearance variations, each time the classifier (model) is updated errors may be introduced into it by imperfect tracking results. The repeated inclusion of these errors over time will most likely lead to degradation of the classifier, also known as the drift problem. The degradation occurs since the classifier gradually loses the ability to distinguish between target and non-target examples. Furthermore, the above-mentioned approaches are online versions of supervised learning techniques; they are not good at tackling noisy labels.

The trade-off between non-adaptive and adaptive classifiers can be portrayed as the stability-plasticity dilemma [Grossberg, 1987]. If a classifier has been trained offline, and it is not updated online then it cannot drift. However, its non-adaptive nature

would not allow robust tracking of a target that is undergoing appearance variations. On the other hand, online classifiers being updated with their own tracking results can easily lead to tracking failure in the case of incorrect updates.

An intuitive way to reduce the drift problem is to bind an online appearance learner (tracking classifier) with a fixed (non-adaptive) initial appearance (auxiliary classifier). One of the first attempts in this direction was made by Grabner et al.[Grabner et al., 2008], in which they used an offline classifier in conjunction with an online classifier. The coupling of an adaptive classifier with a fixed classifier overcomes the drift problem to a certain extent, but does not allow the method to adapt to appearance changes beyond a certain range. The drift problem can also be alleviated by training a pair of independent classifiers. [Yu et al., 2008] adopted a co-training based approach to label the incoming data and train a hybrid generative discriminative model. The generative model encodes all the appearance variations of the target that have been seen while tracking, while the discriminative model focuses on the recent appearance variations of the target. The co-training approach requires the feature sets for the classifiers to be conditionally independent, which is a strict assumption for visual tracking.

Online classifiers (e.g. boosting) or learners being updated on their own tracking result can drift due to the label noise problem. The label noise problem emerges when the bounding boxes of the target do not perfectly overlap with the target [Santner et al., 2010]. If label noise persists over a tracking sequence, the tracker will most likely start to loose lock on the target. An online version of the Multiple Instance Learning (MIL) technique was introduced to tracking by Babenko et al.[Babenko et al., 2009] to overcome the label noise problem. Their central idea was to take patches most likely lying on the target as instances for the positive bag, and instances further away from the target as negatives. With this formulation, the ambiguity in tracking results caused by imperfect alignment of the target bounding box and target is passed onto the learning algorithm, which now has to decide which instance in each positive bag is the most correct. Most correct implies the instance whose bounding box aligns most closely with the target. Zhong et al. [Zhong et al., 2010] handled the label noise problem by framing tracking in a weakly supervised learning scenario where the labels (possibly noisy) for positive and negative examples were provided by multiple imperfect oracles (i.e. trackers). Recently, Hare et al. [Hare et al., 2011] proposed an online structured output support vector machine (SVM) to alleviate the effect of misaligned examples. Instead of learning a classifier, which predicts binary labels, the approach learns a predictor that directly predicts the target transformation between frames.

The drift problem can be explicitly addressed by combining trackers with different appearance adapting capabilities or by integrating a tracker and a detector. Santner et

al.[Santner et al., 2010] heuristically combined an adaptive online learning method with two complementary tracking approaches. A template tracker and an optical-flow-based mean-shift tracker are the non-adaptive and adaptive approaches, respectively, while an online random-forest is a moderately adaptive method. Although the proposed method reduces drift to a reasonable extent, it requires certain thresholds to be adjusted prior to tracking to perform well. Kalal et al. [Kalal et al., 2012] trained a binary classifier by exploiting the structure of unlabeled data during tracking using positive and negative constraints. This binary classifier was used to correct the errors made by the tracker, thereby reducing the drift problem.

The discriminative learning paradigm has also been exploited through mid-level cues, compressed features, and the incorporation of spatial constraints to achieve robust visual tracking. Wang et al. [Wang et al., 2011] proposed a discriminative model based on superpixels to minimize the impact of drift. This appearance model is constructed by clustering segmented superpixels from training images. Rich image representations can be combined with simple discriminative models to build robust and efficient trackers. Based on this formulation, Zhang et al. [Zhang et al., 2012a] projected high-dimensional features, which represent the target and the background samples, to a low-dimensional space using a compressed sensing technique, and then trained a binary classifier on these low-dimensional features to counter appearance variations and achieve computationally efficient tracking. To cope with appearance changes, Zhang and Van der Maaten [Zhang and van der Maaten, 2014] trained appearance models of target parts and the structural constraints between these parts jointly in an online structured SVM.

2.2.3 Other Approaches

Almost all the aforementioned approaches, in both the generative and discriminative learning paradigms, assume that a target model trained (updated) till the previous time-point is still valid for the current time-point. However, the appearance of the target might have changed significantly at the current time-point. As a result, the target model, trained till the previous time-point, would produce a sub-optimal target state at the current time-point, and the errors accumulated over time might cause tracking failure. To address the above problem, Bai and Tang [Bai and Tang, 2012] designed three sets of patches for training the online laplacian ranking SVM. The first and the second set contain patches corresponding to bounding boxes of the target in the initial and the latest frames, respectively, to adapt to the fluctuation in the target appearance. While the third set comprises patches corresponding to rough location of the target from the current time-point to adapt to the substantial appearance variations.

Most of the tracking approaches reviewed in the previous two sections assume that the best target state produces the highest likelihood score near the previously estimated state. This assumption is only valid if the target model, updated over time, is always correct. However, as mentioned in the previous two sections (2.2.1,2.2.2), it is not difficult to imagine that the target model might become contaminated with noise (i.e. background information) in real-world tracking scenarios. This noisy target model would not allow the search method to output the true target state even though it might have converged to the global optimum solution (the best state). The best state found in this case would not correspond to the true state. With this tracking error, the target model includes more background and might drift over time. Recently, Kwon and Lee [Kwon and Lee, 2013] proposed an indirect solution to this drift problem. They employed two Markov Chains in an interactive manner to find the best state as the one that maximizes the average of the lower and upper bounds of the likelihood and at the same time minimizes the gap between two bounds of the likelihood. This best state would produce the same likelihood, regardless of the target models used.

2.3 Search Strategies

Previous section discussed some different methods of modelling target appearance. This section reviews search strategies that use appearance models and image data to infer a solution to the tracking problem. In simple terms, the aim of the search strategy is to find the best hypothesis from the space of all possible hypotheses. The space of all possible hypotheses defines the search space of a search method. The dimensionality of this search space is determined by the number of variables included in the target state multiplied by states per variable.

As mentioned previously, search strategies can be either probabilistic or non-probabilistic. Probabilistic approaches, described in section 2.3.2, use random variables and their corresponding probability distributions to model the uncertainty associated with the motion, observation, state and appearance of the target. These models are then fused to infer knowledge about the target state at each time-step. In contrast, the non-probabilistic approaches discussed in section 2.3.1 employ optimization methods to find the best target state.

2.3.1 Non-probabilistic Search Strategies

Non-probabilistic search strategies typically cast tracking as an optimization problem. With this formulation, the tracking problem is described by a cost function and the solution is achieved by maximizing or minimizing this cost function [Smith, 2007]. These methods gained popularity as they exhibit good convergence properties and are computationally efficient.

One early work in optimization-based tracking is known as EigenTracking [Black and Jepson, 1998]. In this work, the target is represented by a small set of eigen bases and a tracking solution is found by minimizing an error function which defines similarity between the learned bases and the image data through least-squares approximation. Avidan [Avidan, 2004] argued that eigenbasis is not a general purpose classification technique, and proposed a fusion of SVM and an optic-flow based tracker to track vehicle rear-ends. An SVM based detection module was used to find candidate vehicle regions. These detected regions were then tracked by maximizing the SVM classification score using gradient descent.

Another early, but classic optimization technique for tracking is based on the Mean Shift algorithm. Mean Shift is a general non-parametric approach finding the mode of a density function which was introduced by Fukunaga and Hostetler [Fukunaga and Hostetler, 1975]. Later, it was adopted to solve the tracking problem by Commaniciu et al. [Comaniciu et al., 2000]. In [Comaniciu et al., 2000], mean shift iterations are utilized to find the target candidate that best matches the target model. The matching function is a similarity measure based on the Bhattacharyya coefficient.

2.3.2 Probabilistic Search Strategies

Probabilistic search strategies hold three clear advantages over the non-probabilistic methods described in the previous section [Smith, 2007]. The first is flexibility, as one component can be exchanged for another (e.g. an edge based observation model can be replaced with a colour based observation model) without affecting the overall design. The second is their generality: an inference method used for single target tracking might also be applicable to a pose estimation problem. Third, but an important advantage of these methods, is the ability to systematically handle unpredictable target configurations and noise. This is possible because probabilistic methods can maintain multiple hypotheses, which makes them robust to the uncertainties present in real-world data. For instance, these methods are robust to situations in which a target may move unpredictably or, due to clutter, it appears that there may be more than one likely solution.

Recursive Bayesian estimation (RBE) is the most popular and celebrated technique in probabilistic tracking, although there are a few alternatives. RBE formulates tracking as the temporal propagation of conditional densities over time. Let $\mathbf{X}_t$ be the target state at time t, and $\mathbf{Y}_{1:t} = \{\mathbf{y}_1, ..., \mathbf{y}_t\}$ be the observations up to time t. According to RBE, the posterior probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ given the state $\mathbf{X}_t$ at time t and observations $\mathbf{Y}_{1:t}$ up to t can be computed as:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) = C_t p(\mathbf{Y}_t|\mathbf{X}_t) \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) d\mathbf{X}_{t-1}, \tag{2.1}$$

where the term $p(\mathbf{Y}_t|\mathbf{X}_t)$ expresses the observation model. It measures how well the observation $\mathbf{Y}_t$ at time t supports the hypothesis that the target is in state $\mathbf{X}_t$ at time t. $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is called the motion model or the state evolution, and determines the probability of transitioning to state $\mathbf{X}_t$ at time t given state $\mathbf{X}_{t-1}$ at time $t-1$. $C_t = \frac{1}{p(\mathbf{Y}_t|\mathbf{Y}_{1:t-1})}$ ensures that the posterior probability sums to one over the state space.

Eq.2.1 is composed of two steps (See Fig. 2.9): prediction and update. During prediction, the posterior probability at time $t-1$, $p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})$, is propagated to time t using the motion model $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ to form a prior distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t-1})$,

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t-1}) = \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) d\mathbf{X}_{t-1}, \tag{2.2}$$

In the update step, this prior distribution is corrected using the observation model $p(\mathbf{Y}_t|\mathbf{X}_t)$ to generate the posterior probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ at time t.

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) = C_t p(\mathbf{Y}_t|\mathbf{X}_t) p(\mathbf{X}_t|\mathbf{Y}_{1:t-1}), \tag{2.3}$$

The posterior probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ encodes belief about the state of the target at time t using observations $\mathbf{Y}_{1:t}$ up to time t. The next section reviews well-known methods for computing recursive Bayesian estimation.

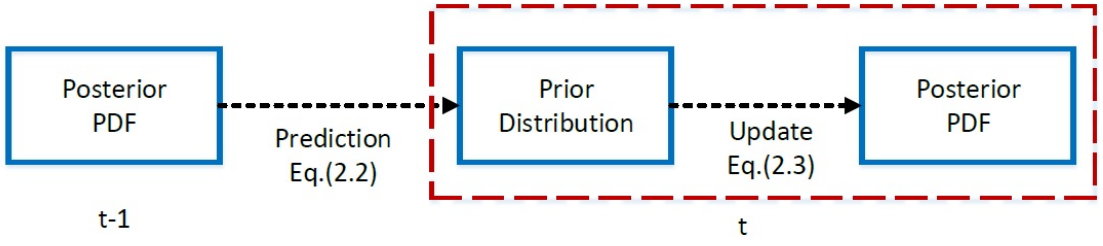


FIGURE 2.9: Steps involved in estimating the posterior PDF at time t from the given posterior PDF at time $t-1$ using RBE.

2.3.2.1 Kalman Filter

The Kalman Filter dates back to 1960, when it was formally presented by R. Kalman [Kalman, 1960]. In general, the Kalman filter provides an optimal way of estimating the hidden state of a system by analyzing observable measurements [Kalman, 1960]. In other words, the Kalman filter solves the state estimation problem completely. It is important, however, to realize that the filter provides an optimal solution only if certain assumptions hold true.

In terms of Bayesian tracking, the first assumption is that the model governing the state evolution must be linear with additive Gaussian noise, and the second assumption is that the observation density must be Gaussian. If these assumptions are valid for a tracking situation then the Kalman Filter will provide an exact solution to the recursive Bayesian estimation.

Informally, the filter can be described as the composition of three stages: deterministic prediction, stochastic diffusion, and prediction correction through measurement. In the deterministic prediction stage, the mean value of the state distribution changes while the covariance remains fixed. To capture inaccuracies in the process of deterministic prediction and the unavoidable noise present in any real-world phenomenon, the stochastic diffusion process is incorporated, resulting in increased state covariance. Finally, this state prediction is corrected using observable measurements to produce a final state estimate. This last step also results in the alteration of the state mean and covariance. In summary, this filtering process is a combination of a prediction step and a correction step. Both deterministic prediction and stochastic diffusion generate the predicted state that is refined using the measurement to estimate the final state.

Mathematically, the predictive component of the Kalman filter, which produces the *a priori* state estimate and its covariance, respectively, can be written in terms of equations 2.4 and 2.5. The *a priori* estimate depicts the predicted state before the arrival of new measurements. The state transition can be written as:

$$\mathbf{X}_t^- = \mathbf{A}\hat{\mathbf{X}}_{t-1}, \tag{2.4}$$

where $\mathbf{X}_t^-$ denotes the predicted state at time t, and $\hat{\mathbf{X}}_{t-1}$ represents the estimated state at time $t-1$. $\mathbf{A}$ is the deterministic motion model that relates the state at time $t-1$ to the state at time t. To reflect the inaccuracy (error) in this state evolution, the covariance of this predicted state is computed according to:

$$\mathbf{P}_t^- = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^T + \mathbf{Q}, \tag{2.5}$$

where $\mathbf{P}_t^-$ is the covariance of the predicted state $\mathbf{X}_t^-$ at time t, and $\mathbf{Q}$ is the covariance of the noise associated with the state prediction process.

The correction component of Kalman filter refines the state prediction by optimally combining it with the measurement. An *a posteriori* estimate of the target state is computed as:

$$\hat{\mathbf{X}}_t = \mathbf{X}_t^- + \mathbf{K}_t\left(\mathbf{Y}_t - \mathbf{H}\mathbf{X}_t^-\right), \tag{2.6}$$

where

$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_t^-\mathbf{H}^T + \mathbf{R}\right)^{-1}, \tag{2.7}$$

and $\mathbf{R}$ is the noise associated with the measurement process. $\mathbf{H}$ is a quantity that maps a state vector into its equivalent measurement vector. Like the predictive component, the *a posteriori* state covariance has to be computed,

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}\right)\mathbf{P}_t^-. \tag{2.8}$$

Thought of as an estimation tool, the Kalman filter has two desirable features: ability to fuse information from different sources in a principled manner, and recursiveness [Cannons, 2008]. Recursivenss implies that at each time-step it does not have to store and reprocess all the previous information.

Despite these useful features, the Kalman filter makes strict assumptions about the nature of system dynamics and observations which typically do not hold in general tracking conditions. For instance, the likelihood densities arising from image features are usually multimodal. As a result, the solution to Eq. 2.1 would be intractable. Under these situations, Particle Filtering is a popular state estimation tool that overcomes the limitations of the Kalman Filter. The following section will describe how particle filtering implements recursive Bayesian estimation.

2.3.2.2 Particle Filters

Particle filters (PF) belong to the class of sequential Monte Carlo (SMC) filters. They can be used to represent the propagation of conditional densities when the dynamics and observation densities involved in a system are non-Gaussian. The core idea of PF is to represent the posterior probability density function (pdf) of a state by a set of weighted particles.

As a state estimation tool, PF can approximate the Bayesian filtering distribution (Equation 2.1). Now, in terms of PF, the prediction and update steps of Bayesian filtering become the propagation and weighting of particles, respectively. To achieve them, dynamic and observation models are applied individually to each of the particles.

Formally, the posterior pdf $p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})$ at time $t-1$ is approximated by a set of weighted particles $\left\{\mathbf{X}_{t-1}^{(i)}, \omega_{t-1}^{(i)}\right\}, i = 1, ..., n$, where $\mathbf{X}_{t-1}^{(i)}$ is the ith particle and $\omega_{t-1}^{(i)}$ is its corresponding weight (Fig. 2.10(a)). Given this representation, the three steps to approximate the posterior pdf at time t are: resampling, propagation of particles, and weighting of particles.

Resampling A common problem with PF is degeneracy, in which one particle dominates the rest after a few iterations. This effect can be reduced by introducing a resampling step, which would eliminate particles with lower weights and concentrate on particles with higher weights. In the resampling step, the weighted particle set $\left\{\mathbf{X}_{t-1}^{(i)}, \omega_{t-1}^{(i)}\right\}$ is resampled (with replacement) according to the weights $\omega_{t-1}^{(i)}$ to get the unweighted particle set $\left\{\mathbf{X}_{t-1}^{(i)}, \frac{1}{n}\right\}$. Fig. 2.10(b) illustrates the result of the resampling step.

Propagation of particles This step is also known as hypothesis generation. Each member (particle) of this set $\left\{\mathbf{X}_{t-1}^{(i)}, \frac{1}{n}\right\}$ is propagated using the motion model $p\left(\mathbf{X}_t|\mathbf{X}_{t-1}\right)$ to approximate the prior distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t-1})$ at time t by an unweighted particle set $\left\{\mathbf{X}_t^{(i)}, \frac{1}{n}\right\}$ (Fig. 2.10(c)).

Weighting of particles This step is also termed hypothesis correction. Here, each particle belonging to the set $\left\{\mathbf{X}_t^{(i)}, \frac{1}{n}\right\}$ is weighted based on the observation model $p(\mathbf{Y}_t|\mathbf{X}_t)$ to approximate the posterior pdf $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ at time t by a weighted particle set $\left\{\mathbf{X}_t^{(i)}, \omega_t^{(i)}\right\}$. Fig. 2.10(d) shows high weight particles with darker tonalities of grey and vice versa as the result of applying the observation model and the approximated posterior pdf at time t is sketched in Fig. 2.10(e).

In general, PF can provide good approximation to the posterior pdf in a computationally efficient way as long as the chosen proposal distribution generates samples near the modes of the posterior pdf. However, when the dimension of the state space increases, as is typically found in multi-object tracking, it becomes difficult to find a good proposal

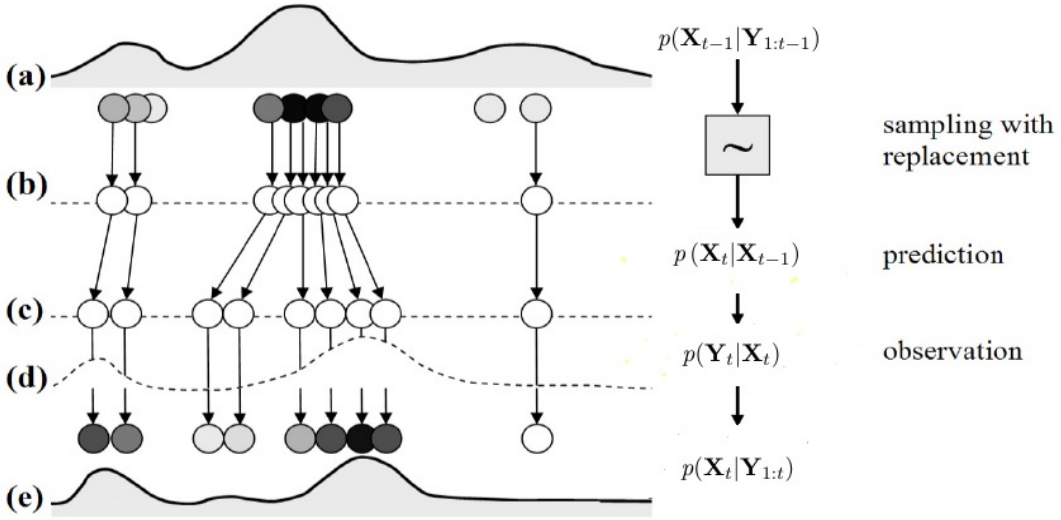


FIGURE 2.10: **Steps involved in one iteration of a particle filter algorithm.** (a) Posterior pdf at time $t-1$ approximated by a particle set is the input to the filter at time t. (b) Resampling step: particles with lower weights are eliminated while particle with relatively higher weights are selected multiple times. (c) Propagation of particles: Each particle in the unweighted particle set is propagated using the motion model to approximate the prior distribution at time t. (d) Weighting of particles: Propagated particles are weighted according to the observation model. (e) These weighted particles approximate the posterior pdf at time t. This figure is taken from [Smith, 2007].

distribution. As a result, the efficiency of PF decreases as it requires an exponentially large number of samples to cover this high-dimensional space. To improve the efficiency of SMC filters, methods based on Markov Chain Monte Carlo (MCMC) were proposed. The following section will describe one of the most popular and widely used MCMC method in visual tracking and a few advanced MCMC methods.

2.3.2.3 Markov Chain Monte Carlo

MCMC sampling methods construct a Markov Chain whose implicit stationary state probability approximates the target posterior. In this formulation, structural knowledge of the state space can be incorporated into the sampling, and the particles are generated more frequently in important regions of the underlying posterior, which improves efficiency.

Among many MCMC sampling methods, Metropolis-Hastings (MH) is the most famous and has been widely applied to visual tracking problems [Khan et al., 2005],[Smith et al., 2005],[Kwon and Lee, 2010]. The MH algorithm constructs a set of particles to approximate the Bayesian filtering distribution (Eq. 2.1). In contrast to the previously described sampling method, PF, this algorithm approximates the posterior pdf by an unweighted particle set $\left\{\mathbf{X}^{(i)}, \frac{1}{n}\right\}, i = 1, ..., n$. The following subparagraph will describe

how the MH algorithm is used to define the Markov Chain whose stationary distribution $\pi(\mathbf{X})$ will be equal to the sought posterior pdf $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$.

The Metropolis Hastings (MH) Algorithm Metropolis Hastings performs a random-walk to approximate the stationary distribution $\pi(\mathbf{X})$. Each move of the algorithm proposes a state, which is accepted or rejected using an evaluation criterion.

Let $\mathbf{X}_t^{'}$ be the state proposed by the proposal density $Q(\mathbf{X}_t^{'}|\mathbf{X}_t^q)$ in the q_{th} iteration of the MH algorithm. $\mathbf{X}_t^{'}$ is then evaluated using the acceptance ratio a,

$$a = min\left[1, \frac{\pi(\mathbf{X}_t^{'})Q(\mathbf{X}_t;\mathbf{X}_t^{'})}{\pi(\mathbf{X}_t)Q(\mathbf{X}_t^{'};\mathbf{X}_t)}\right]. \tag{2.9}$$

If $a \geq 1$, the proposal $\mathbf{X}_t^{'}$ is accepted and added to the Markov Chain $\mathbf{X}_t^{q+1} = \mathbf{X}_t^{'}$. Otherwise, the proposal is accepted with probability a. If the proposal is not accepted, then the previous state is added to the Markov Chain $\mathbf{X}_t^{q+1} = \mathbf{X}_t^q$.

Although the steps required to simulate the Markov Chain using the MH algorithm are quite straightforward, quick convergence of Markov Chain to the underlying invariant distribution might not be an easy task. To make sure that it converges in a reasonable time, a carefully designed proposal density is required. For instance, if the variance of this proposal is too small, then the chain might visit only a few of the posterior modes. On the other hand, a higher variance in this proposal may cause a chain to get stuck in one of the posterior modes for a long time, due to a high rejection rate. The negative impact of mistuned variance in the MH algorithm becomes especially pronounced in large search spaces. In what follows, some alternative MCMC sampling methods, which attempt to overcome the aforementioned limitations of the MH algorithm, will be discussed briefly.

Adaptive MCMC algorithms Adaptive MCMC algorithms [Roberts and Rosenthal, 2009] provide an automatic way of tuning the proposal variance, in order to maintain a certain acceptance rate of the sampler, and thus can better mix the different modes of the posterior pdf. The proposal variance is tuned on-the-fly to produce an acceptance rate close to the optimal value 0.44 [Roberts et al., 1997]. Roberts et al. [Roberts et al., 1997] showed that high dimensional target distributions of dimensionality d satisfying certain moment conditions, the Optimal value of the proposal variance $\hat{\lambda}_d$ satisfies $d^{0.5}\hat{\lambda}_d = l$, for some fixed l which is contingent on the roughness of the target distribution. This optimal proposal variance then leads to the optimal acceptance rate of 0.44. For more details on the theoretical derivation of this optimal value, please refer

to [Roberts et al., 1997],[Rosenthal et al., 2011], and [Sherlock, 2006]. The capability to automatically tune the proposal variance makes adaptive methods effective when required to visit large areas of the search space, e.g. when tracking abrupt target motion. However, the method lacks a systematic way of escaping local maxima, and an efficient sampling scheme to deal with large state spaces.

Partition-based algorithms These methods divide state space into a number of equally sized cells, and use exploration and exploitation moves to approximate the complex target distribution. The exploration moves encourage the method to visit cells that have not been explored enough, while the exploitation moves let the method spend more time in cells that contain modes of the target distribution.

Kwon and Lee [Kwon and Lee, 2008] combined the Wang-Landau Monte Carlo (WLMC) method with the MCMC method to escape local maxima in a complex posterior distribution, while searching in a regular grid that divides the image space in a number of equally sized cells. In [Kwon and Lee, 2008], the Wang-Landau method estimates the Density of States (DOS) term, which denotes the extent to which cells have been explored, and this term is used to generate moves to cells that have not been explored enough. This allows discovery of local maxima in specific cells, while jumping between them. The likelihood term in MCMC causes this method to spend more time in cells that contain highly probable target states. With this term, the method expends more samples around the current local maximum, which has already been well explored. This thesis generalizes this search method to cells of variable size and location in Chapter 5.

Along similar lines, [Zhou et al., 2012] introduced Stochastic approximation Monte Carlo (SAMC) sampling into the Bayesian filtering to approximate the target distribution, while searching in a regular grid. The method generates global and local moves to locate possible modes and exploit local mode structure, respectively.

To summarize, in the beginning of this chapter, major visual tracking problems were described. Then a few basic as well as some advanced approaches to appearance modelling with a greater emphasis on the recent state-of-the-art were reviewed. Finally, classic search methods that use appearance models to find the optimum solution to the tracking problem were touched upon.

Before delving into the in-depth discussion about approaches addressing occlusions and abrupt motion variations, the following section briefly describes a few tracking methods based on object detection and data association paradigm. It is a popular model for tracking multiple targets [Zhang et al., 2015] and some of the approaches based on this paradigm are related to the work presented in this thesis.

2.4 Tracking by detection and data association

Recent progress in object detection research [Felzenszwalb et al., 2010, Wu and Nevatia, 2007, Zhu et al., 2006] has produced many multi-target tracking frameworks based on object detection followed by data association process. An offline or an online learned object detector is used to find candidate locations of targets in each frame of the sequence. Then, a data association process aims to estimate the tracks of the targets by linking detection responses corresponding to same target over time. However, the detector can output missed detections (false negatives), false alarms (false positives), and inaccurate detections. The complexity of the association process increases which now has to handle these problems along with the linking task (assigning a unique detection to a track) [Zhang et al., 2015].

Local Linking-based Methods To overcome the association problems, some works [Cai et al., 2006, Khan et al., 2005, Li et al., 2008, Okuma et al., 2004, Wu and Nevatia, 2007] have tried to solve the correspondence (linking) problem locally i.e. using information from a single or multiple close by frames. [Wu and Nevatia, 2007] associated detections frame-by-frame by proposing a similarity measure for matching detections based on cues from motion, size and colour and used a greedy algorithm to establish correspondence between tracks and detections. [Okuma et al., 2004] proposed a combination of mixture particle filter and Adaboost detector to track multiple players frame-by-frame. The proposal distribution is constructed by a mixture model that merges information from the motion models of the players and the Adaboost detection. The learned Adaboost detector allows to quickly detect the new players entering the field of view while the autoregressive dynamics lets tracking of individual players. Local association-based methods are prone to tracking failure when the targets occlude each other as the noisy detections complicate the data association [Huang et al., 2013].

Global Linking-based Methods In contrast to local approaches, many efforts have been made to solve the association problem for multiple target tracking by looking over a longer time window. Multiple Hypotheses Tracking (MHT) [Reid, 1979] and Joint Probabilistic Data Association Filter (JPDAF) [Fortmann et al., 1983] are among the earliest to resolve the correspondence problem in multi-target tracking. These approaches maintain multiple hypotheses until enough information can be collected to resolve the ambiguity [Huang et al., 2013]. These methods are computationally expensive since search space grows exponentially with the number of frames. To mitigate this difficulty, some works have tried to simultaneously optimize all trajectories by using different optimization techniques. For instance, [Andriluka et al., 2008] used Viterbi algorithm to recover

target paths, and [Kaucic et al., 2005, Srinivas et al., 2006] used Hungarian algorithm to simultaneously find the optimal trajectories.

Hierarchal Methods Some recent works have combined local linking with global association to solve the multi-target tracking problem. These approaches generate tracklets (short tracks) by connecting detections between neighbouring frames and then progressively link tracklets into longer tracks using global association methods. For example, [Xing et al., 2009] proposed a detection-based two-stage framework for multi-target tracking. In local stage, a particle filter is used to generate reliable tracklets. In the global stage, detection responses are collected from a temporal sliding window to generate a set of potential tracklets. The reliable tracklets generated from the local stage and a set of potential tracklets collected from temporal window are associated by Hungarian algorithm to get the global association. [Huang et al., 2013] presented a hierarchal association framework in which a two-threshold conservative strategy was used to link detection responses in consecutive frames to form tracklets and a Hungarian algorithm was used at the higher stages of the hierarchy.

In what follows, approaches handling two outstanding problems in visual tracking, occlusion and target motion variations will be re-visited, since they are relevant to the work described in this thesis.

2.5 Existing Approaches for Handling Occlusion

As described in section 2.1.3, occlusion is a classic problem in visual tracking, and a number of approaches have tried to address it. Generally, occlusion handling may be explicit or implicit. Explicit approaches depend on some detection mechanism to reveal occlusion events, while implicit approaches employ rich target representations that are robust to occlusion. A few approaches scan the whole image or use context around the target for re-detection after occlusions.

Explicit Approaches Explicit occlusion handling requires robust occlusion detection. Yin et al. [Yin and Collins, 2008] presented a combination of local and global mode seeking techniques. During normal tracking, when the target is visible from frame to frame, local optimization was used to track the local mode of the objective function whose arguments are translation, scale, and rotation parameters of the target bounding box. To recover from occlusions, adaptive simulated annealing (ASA) was adopted as a global optimization technique with the same objective function to detect the object through stochastically sampling the large 4D space of translation, scale, and rotation.

Occlusion detection was achieved with a naive threshold based on the value of the objective function used in local mode seeking. Along similar lines, Avidan [Avidan, 2007] detected occlusions by setting a threshold on the classification score.

Lerdsudwichai et al.[Lerdsudwichai et al., 2005] detected occlusions by using an occlusion grid with a drop in similarity value. The occlusion grid records locations that objects occupy in an image, it is same size as the video frame and each cell of the grid represents a pixel of the video frame. The similarity value is the distance between the tracked object model and the image observation corresponding to the most likely object configuration. Prior to tracking, each cell in this grid is initialized with the number of the object occupying the corresponding pixel in the first frame, and, while tracking, this grid is updated with the tracking results. Occlusion between objects is detected by searching the cells of this grid, and variations in similarity values of objects are examined to find the occluder and the occluding object. This approach can produce false alarms because the required drop in similarity could occur due to natural appearance variation.

To explicitly tackle occlusions, Kwak et al. [Kwak et al., 2011] trained a classifier on the patterns of observation likelihoods in a completely offline manner. In this approach, a target is divided into regular grid cells, and the classifier is used to determine the occlusion status of each cell. However, training and testing must be performed in the same environment to ensure the reliability of this algorithm, and the tracking algorithms used (for training and testing) should be identical.

In [Mei et al., 2011] and [Bao et al., 2012], an occlusion map is generated by examining trivial coefficients, and is used to determine the occlusion state of a tracking result. Trivial coefficients correspond to trivial templates that are used in a sparse representation to account for image corruptions such as occlusions. These get activated (become non-zero) when the pixel intensity in a given image observation cannot be well approximated by the target templates. The occlusion map is processed by applying morphological operations to approximate the occluding region, and if the area of this region is greater than a pre-defined threshold, the corresponding template is said to be occluded. Both these methods are prone to false positives when it is hard to separate the intensity of the occluding object from small random noise on the occluded object. On a general note, from the above discussion, it can be observed that reliable occlusion detection is an open problem.

Implicit Approaches Implicit approaches can be divided into two categories. The first is based on adaptive appearance models which use statistical analysis [Han and Davis, 2005, Jepson et al., 2001, Ross et al., 2008] to reason about occlusion. In [Jepson et al., 2001], the target model is described by a mixture of 3 components, which

is updated online to account for appearance variations. In [Han and Davis, 2005], a more flexible target model based on a mixture of Gaussians was proposed, in which the number of Gaussians was automatically determined according to the temporal appearance variations. Although adaptive appearance models can improve tracking accuracy, they expect noise-free appearances for learning. During partial and full occlusions, the state estimates of the tracker can be poor. Passing the noisy appearances corresponding to these poor states to an online learning method would quite likely result in an inappropriate appearance model being learned.

Approaches in the second category divide the target into patches to indirectly reason about occlusion using robust statistics [Adam et al., 2006], [Han and Davis, 2009]. [Adam et al., 2006] preserves the spatial distribution of pixel intensities with a patch-based target representation. Every patch votes on the hypothesized positions and scales of the target at each moment in time during tracking. Outliers (occluded patches) are rejected by applying an outlier detection mechanism on the voting maps. [Han and Davis, 2009] tracks parts in a low dimension parameter space, and estimates the high dimensional parameters by statistically combining the individual tracking results. These approaches are, however, susceptible to failure in case of full occlusions. For instance, in [Adam et al., 2006], when the number of occluded patches increases beyond a certain percentage, the tracker can jump to a non-target region.

Some recent works have deployed multiple appearances in different formulations to indirectly counter occlusions. Kwon and Lee. [Kwon and Lee, 2010] selected a fixed number of target models for a sampling based search method. To improve the efficiency and the accuracy of the sampling process, [Kwon and Lee, 2011] sampled target models using the recent tracking history. Recently, Park et al. [Park et al., 2012] modelled the probabilistic dependency between sequential target appearances, in order to infer the most probable target appearance at the current time-step. Though effective in handling various appearance variations and short-term occlusions, these methods can still lose the target during longer-term occlusions as they do not use temporal information other than from the immediate previous time-step to estimate the state at the current time-step.

Approaches Exploiting Detectors and Context It is possible to re-acquire a target after occlusion by scanning the whole image space with a target model (classifier). A few approaches have exploited detectors to re-locate the target after occlusion [Kalal et al., 2012], [Grabner et al., 2008]. However, the detector could report false positives in the presence of distractors, causing the tracker to fail. Moreover, these approaches are not computationally efficient as they search the whole image space once the target is lost.

Recently, a class of tracking methods which not only take the target description into account but also consider its context have shown improved robustness in overcoming occlusions. Grabner et al. [Grabner et al., 2010] and Yang et al. [Yang et al., 2009b] employed spatio-temporal context to learn the target location while it is occluded. In particular, [Grabner et al., 2010] determines the location of the target when it is invisible by introducing supporters, which are local image features that bear a strong or weak temporal correlation to the motion of the target. In a similar pursuit, Dinh et al. [Dinh et al., 2011] developed a new tracking framework based on supporters and distractors. Distractors are the regions that have similar appearance to the target, while supporters correspond to local image features around the target whose motion is statistically related to the target over a short temporal scale. Although these approaches are robust to occlusions, they rely on the presence of auxiliary objects to re-acquire the target after occlusions.

Other Approaches Some approaches address domain-specific occlusion of known target types. Lim et al. [Lim et al., 2006] proposed a human tracking system based on learning dynamic appearance and motion models. In [Lim et al., 2006], dynamical appearance and motion models are learned from a small set of initial frames using robust system identification techniques. Occlusion detection is achieved by setting a threshold on the likelihood value, which can report a false alarm in case of an appearance variation. A three-dimensional geometric hand model was proposed by Sudderth et al. [Sudderth et al., 2004] to reason about occlusion in a non-parametric belief propagation tracking framework. This approach requires a detailed model of the desired target to be built before reasoning about occlusion, which is not always possible.

Occlusions can be dealt with using a multi-camera setup. [Dockstader and Tekalp, 2001] used a Bayesian belief network to fuse inputs from multiple views, and [Fleuret et al., 2008] combined a probabilistic occupancy map, generated from multiple frames at each time instant, with a global optimization to overcome occlusion. As most videos are shot with a single camera, and multiple cameras bring additional costs, this is not a generally applicable solution.

Another method of handling occlusion is to alter the search mechanism of the tracking process. Arnaud and Memin [Arnaud and Mémin, 2007] dealt with occlusions by modifying the diffusion process in the particle filter using partial linear gaussian models. These models allow the use of optimal importance function for the diffusion process, and thus, the resulting algorithm explores the state space in an optimal way. Karavasilis et al. [Karavasilis et al., 2011] tackled occlusion by forwarding the estimated target location to a Kalman filter whose parameters are determined online based on recent motion history.

Although these methods may be effective in dealing with occlusions, they tend to be computationally expensive. For instance, in [Arnaud and Mémin, 2007], the image-based motion model used to propagate each particle costs more than the usual auto-regressive motion models.

2.6 Existing Approaches for Handling Motion Variations

In 2D visual tracking, target motion can exhibit many variations. In general, these variations can be classified as either smooth or abrupt over some time interval. To facilitate efficient tracking, most tracking methods assume that the target motion varies smoothly. However, in real-world scenarios, the target motion can undergo abrupt variations due to its own unexpected movement, camera switching/motion, and low-frame rate image acquisition. Furthermore, a target can accelerate and decelerate, and this motion can be thought of lying somewhere between the boundaries of smoothly and abruptly varying motion.

An important component of Bayesian tracking (Eq. 2.1) is a motion model, which describes the expected motion of a target over time. Motion models can guide the search towards the correct modes of the target distribution. In other words, the search space of tracking parameters to be estimated can be reduced. Consequently, the search method would be computationally efficient and immune to local optimum. In what follows, motion modelling techniques and approaches without motion prior will be reviewed.

General-Purpose Motion Models Since it is difficult to produce an accurate motion model for a large variety of tracking environments, sampling-based tracking frameworks have conventionally depended on a single general-purpose motion model, with parameters set a-priori, like Random Walk (RW) [Chang and Ansari, 2005],[Perez et al., 2004] or Nearly Constant Velocity (NCV) [Shan et al., 2007],[Pernkopf, 2008]. The RW model is based on the assumption that the target's velocity is a white noise sequence, and therefore, it is temporally totally uncorrelated. In contrast, the NCV model assumes that the target's velocity is fully correlated, and changes in this velocity happen due to white noise of acceleration. In short, the two models lie at the two extremes in terms of how much the velocity is temporally correlated [Kristan et al., 2010].

A drawback of these general-purpose models is their inability to handle complex motion variations. For instance, if the target accelerates or suddenly changes its motion direction, these models cannot deliver accurate tracking.

It is possible to cover large motion uncertainty with these models by increasing the process noise. Stochastic search methods such as particle filters [Isard and Blake, 1998a] usually require a larger number of particles when process noise is kept high, to cover the unmodelled dynamics of the target. As a result, the tracking task becomes computationally expensive and the search method becomes vulnerable to distractors.

Hybrid Approaches To overcome the limitations of general-purpose motion models embedded in particle filters (PF), some works have proposed a hybrid of particle filter and Mean Shift algorithms. Maggio and Cavallaro [Maggio and Cavallaro, 2005] formulated a hybrid particle filter/Mean Shift tracker with an adaptive transition model to track abrupt target motion with fewer particles. Particles were generated using a zero-order model with adaptive Gaussian noise, and then Mean Shift (MS) was independently applied to each particle to find local modes of the posterior. However, if the PF tends towards a local maximum the MS step will accelerate the process. To alleviate this effect, Naeem et al.[Naeem et al., 2007] combined Annealed Particle Filter (APF) with Mean Shift. The annealing process in PF smoothes out the likelihood function, which makes the global maximum clearer and allows particles to spread further through increased process noise. The algorithm is robust to local clutter as the MS applied at each stage of annealing pulls particles towards the true target, but is computationally expensive as it requires MS to be applied to each particle at each step of the annealing process.

Improved Proposal Distributions One approach to the increased variance in estimation caused by high process noise is to make an efficient and informed proposal distribution. Okuma et al. [Okuma et al., 2004] designed a proposal distribution that mixed hypotheses generated by an AdaBoost detector and a standard autoregressive motion model to guide a particle filter based tracker. The combination of two approaches allowed their framework to detect targets entering the scene and at the same time achieve consistent track formation. Kristan et al. [Kristan et al., 2010] formulated a two-stage dynamic model to improve the accuracy and efficiency of bootstrap particle filters. The two-stage dynamic model is comprised of a liberal and a conservative model. The liberal model is responsible for greater perturbations in the target's dynamics, while the conservative model assumes smaller perturbation in the target's dynamics. The method fails when the target exhibits frequent spells of non-constant motion. Kwon and Lee [Kwon and Lee, 2011] utilized the recent sampling history to generate candidate proposal distributions (motion models) which were then sampled randomly using Reversible Jump Markov Chain Monte Carlo (RJMCMC). The approach requires the recent sampling history to be clustered and evaluates each tracker sample using the recent tracking history, which might be cumbersome for even a near real-time tracking application. To

handle complex target motion, Mikami et al. [Mikami et al., 2009] used the entire history of estimated states to generate a prior distribution over the target state at immediate and some future time-steps. The accuracy of the prior distribution relies on two strict assumptions: similar target states repeat often and, if the current state is similar to the retrieved past state, the temporal development of current and past states will be similar.

Offline Learned Motion Models Several attempts have been made to learn motion models offline. Isard and Blake [Isard and Blake, 1998b] learned a small set of motion models from ground truth data, and used a hardcoded finite state machine (FSM) to manage transitions between them. Later, North et al. [North et al., 2000] extended the work of [Isard and Blake, 1998b] by learning the parameters of more complex dynamics. They illustrated the effectiveness of their approach on a juggling example by jointly learning the parameters of each motion class and the transition probabilities among these classes. Madrigal et al. [Madrigal et al., 2012] guided a particle filter based target tracker with a motion model learned offline. This model was based on the local motion observed by the camera. Pavlovic et al. [Pavlovic et al., 2000] learned models of human dynamics from motion capture data. An obvious limitation of offline learning is that models can only be used to track the specific class of targets for which they are trained.

Buchanan and Fitzgibbon [Buchanan and Fitzgibbon, 2007] derived a robust motion prior from the global motion of 2D feature points in a temporal window, and then used this prior in the conventional Bayesian framework to track these points. Cifuentes et al. [Cifuentes et al., 2012] developed a few specialist motion models, which only track well in scenes having similar geometry and camera motion, and proposed a classification based approach to automatically predict the right specialist motion model from a set of available models for a given video sequence. Though these specialist models are better than the general-purpose models under known conditions, their design and the training of the selection method, which chooses the most suitable one, requires supervised discovery of motion classes beforehand.

Approaches Without Motion Prior Abrupt or discontinuous motion is difficult for any motion model to capture. Early solutions to this problem were based on a hierarchical search strategy, in which the search proceeds from coarse (large) to fine (small) scales in the spatial domain. The intuition behind this strategy is that the search results in coarse scales may be refined by those in finer scales. An obvious advantage of this approach is the computational efficiency, since it is required to search a large state space. Sullivan et al. [Sullivan et al., 1999] mingled image observations from multiple scales to perform efficient search at the fine scale. A potential drawback

of this approach is that the propagation of error from the coarse scale might cause the search results in the finer scale to deviate from the true optimum. To solve this error propagation problem, Hua and Wu [Hua and Wu, 2004] proposed a framework based on a dynamic Markov network to search collaboratively in different scales, and developed a sequential belief propagation scheme to perform inference in this dynamic model. While this approach alleviates the effect of error propagation mentioned earlier, the risk of losing image information by down-scaling cannot be ignored. Recently, Li et al. [Li et al., 2008] combined observers learned at different temporal scales with a cascaded particle filter to deal with large motion uncertainty. Observers have different discriminative powers, and are utilized at designated stages of the cascade. While this approach has shown promising results in tracking faces, it requires offline training prior to tracking, and thus it cannot be readily applied to track any object.

Abrupt motion can be countered by combining local sampling methods with global sampling schemes. The reason why local and global sampling methods are combined is as follows. Local samplers can trap in local optimum (local-trap problem) when the energy landscape of the filtering distribution is rough as in cases of abrupt motion. Whereas global samplers have the ability to jump between the different possible modes of the target distribution under these situations, but they are not good at exploring local mode structure. Their integration produces the possibility of avoiding the local-trap problem and yet exploring the local mode structure while searching large state spaces. Kwon and Lee [Kwon and Lee, 2008] combined an efficient sampling method with an annealing procedure to search the whole image after dividing it into a fixed grid of equal sized cells. Towards a similar goal, Zhou at al. [Zhou et al., 2012] introduced a new sampling method, known as Stochastic Approximation Monte Carlo (SAMC), into the Bayesian tracking to search for the optimal target state in a regular grid.

Some works have employed more general graphical models than the typical first-order Markov Chain used in probabilistic tracking along with patch matching techniques [Korman and Avidan, 2011] to cope with motion discontinuities. Recently, Hong et al. [Hong et al., 2013] proposed an offline tracker that selects easy-to-track frames first out of the remaining ones and delays tracking of troublesome frames till the end. To track a frame, posterior density from already tracked frames is propagated to this frame in a recursive manner. The density propagation is implemented by a patch matching technique. Though this approach has shown improved tracking in case of abrupt motions and other challenges, it requires and strongly relies on an efficient and a robust patch matching technique for the density propagation. Furthermore, this approach completely discards the temporal coherency information of the target, which might be helpful in resolving appearance ambiguities caused by distractors. Along similar lines, [Hong and Han, 2014] proposed a more accurate offline tracker that addresses the tracking problem through

learning an optimal tree structure in a given video. The idea is to find an optimal tree structure appropriate for tracking, and then improve performance by making use of the found structure. To solve both these problems jointly, an iterative framework based on MCMC technique is employed. A new tree structure is proposed by sampling and validated by tracking in each iteration.

2.7 Observations

This section will underscore a few general as well as some specific observations about the existing approaches to occlusion and motion variations that motivated the research described in the later chapters of this thesis.

2.7.1 Regarding Current Solutions to Occlusion

Many approaches have used complex appearance models to implicitly address partial occlusions and short-term full occlusions [Adam et al., 2006, Han and Davis, 2005, Kwon and Lee, 2010, 2011, Ross et al., 2008]. However, during long-term full occlusions it is quite difficult for any appearance model to stay valid no matter how complex it is. From the computational perspective, such appearance models are usually not feasible in real-time tracking applications. So, a relatively simple appearance model along with the availability of reliable motion information might be a better approach to recovering from long-term full occlusions.

Reliable occlusion detection is an unsolved problem in the visual tracking community. This is because often the functions used for explicit occlusion identification can easily report false alarms when a target's appearance changes abruptly. A tracking algorithm based on implicit occlusion handling that treats the two tracking modes, tracking without occlusions and tracking with occlusions, in a similar manner can be a plausible solution. Since such an approach would not distinguish between the two tracking modes, it may provide a well-grounded recovery after occlusions.

An exhaustive search over the whole image space might re-capture the target after a full occlusion, but it is vulnerable to distractors and often computationally infeasible. Moreover, approaches based on this strategy require reliable occlusion detection to trigger the exhaustive search, whose flaws have been described earlier. Instead of this brute-force search over the whole state space, motion information from multiple previous time-steps can trigger a guided search around the likely modes of the posterior only.

The exploitation of context, making use of auxiliary objects around the tracked target, has shown some success in handling full occlusions [Dinh et al., 2011, Grabner et al., 2010, Xiong et al., 2012, Yang et al., 2009b]. However, these approaches are mainly based on the assumption that either a rigid or non-rigid context is available around the target throughout tracking.

Just as auxiliary objects provide contextual information which helps overcome occlusion, this thesis suggests that models built and used over multiple temporal scales can aid target recovery when it re-appears.

A common, but important feature of approaches to occlusion handling is that they work on a single temporal scale i.e. $[t-1, t]$. Based on a first-order Markov Chain assumption, they infer a target's state at time t using only its state at time $t-1$. The underlying assumption for single scale trackers is that the length of occlusion is shorter than the temporal scale used. However, this is a strict assumption since occlusions can occur over different periods of time. A single scale tracker might not be able to reliably handle variability in the period of occlusion.

Contrary to tracking over single scale, this thesis proposes visual tracking over multiple temporal scales to implicitly handle occlusions of variable lengths (Chapter 4) without requiring complex appearance models and exhaustive search methods.

2.7.2 Regarding Current Solutions to Motion Variations

The design of an informative proposal distribution can help sampling-based trackers to counter complex target motion. These proposals are often based on hypotheses generated by an offline trained object detector. Because the detectors scan the whole image, they could produce false positives in the presence of distractors. Such false positives can lead to drift, if the appearance model used in the tracker cannot discriminate the tracked target from visually similar objects.

Multiple motion model approaches have been used to deal with a range of target motion. Approaches based on offline learned motion models can represent various motion classes effectively, but they can only be used to track targets whose motion is learned prior to the start of tracking. Furthermore, the offline learning requires ground truth on target behaviour, which might be difficult to obtain for very long image sequences. Due to the limited applicability of these approaches, interest towards online development of multiple motion models has grown in the recent past. As reported in the literature, these multiple motion models are either RW models with different variances, which are learned from the recent sampling history to perform MCMC based inference, or a set

of specialist models in which each member is responsible for handling a certain type of target motion.

This thesis addresses a similar challenge, that is to capture reasonable variations in the target's path, but from a temporal perspective. Instead of using a set of specialist motion models, a set of motion models is learned from the target history at multiple temporal scales. This learning over multiple temporal scales lets the proposed tracker maintain a richer description of the target's path. To show the potential advantages of learning over multiple temporal scales, not only motion but appearance variations as well are studied in the ground truth data of several image sequences in Chapter 3.

The existence of multiple competing hypotheses or predictions by the multiple motion models opens up the possibility of a wider range of search methods. To search for the target in a fixed grid of equal sized cells, an integration of the Wang-Landau method and the Markov Chain Monte Carlo (MCMC) method has recently been introduced [Kwon and Lee, 2008]. Chapter 5 generalizes this search method to cells of variable size and location, where the cells are formed around the predictions generated by multiple motion models.

Chapter 3

Potential Benefits of Multiple Temporal Scales

Most visual domain trackers interact with the estimated states over time at a single, fixed temporal scale, usually $[t-1;t]$ [Huttenlocher et al., 1993],[Murray and Basu, 1994],[Dellaert et al., 1999], [Koller-Meier and Ade, 2001],[Matthews et al., 2004],[Khan et al., 2005], [Wu and Nevatia, 2006],[Adam et al., 2006],[Wu and Nevatia, 2007],[Ross et al., 2008], [Moreno-Noguer et al., 2008],[Yang et al., 2009a],[Mei and Ling, 2009],[Kwon and Lee, 2009],[Kwon and Lee, 2010],[Babenko et al., 2011],[Jia et al., 2012],[Park et al., 2012], and [Xing et al., 2013]. In other words, they typically learn a model (be it appearance or motion) at a single temporal scale utilizing only the most recent state history, and use this model over a single temporal scale in the future, e.g. $[t;t+1]$, to localize the target. In contrast, this thesis introduces the idea of visual tracking over multiple temporal scales. This implies that more than one model could be used to estimate the target state at each moment in time. Each model is learned at a given temporal scale using a particular history of tracked states and applied over multiple temporal scales in the future. Now, this formulation raises a three-fold question. **(1)** if a larger set of models is available, are any of these models able to make better predictions than the standard first order Markov model using only $t-1$? A slightly different, but related question would be is there any benefit (in terms of predictive ability) to be gained by switching between these models? **(2)** Can we utilize the models produced from multiple state histories to predict multiple time-steps ahead e.g. $[t;t+1]$, $[t;t+2]$, or $[t;t+6]$ and so deal with variable length occlusions? **(3)** How can we select the most suitable model from a set of models and integrate this selection process into a tracking framework?

This chapter addresses the first question only i.e. by creating a set of models (be it motion or appearance) derived from different temporal scales, can better prediction performance be generated, than would be possible if only a single model from this set were available. The other two questions will be addressed in chapter 4.

This chapter begins by emphasizing the role of multiple scales in the *spatial* domain to build motivation for multiple *temporal* scales in visual tracking (Section 3.1). Then, it investigates both motion and appearance variations in the ground truth data of several image sequences to show the potential benefits of access to a set of models extracted from multiple temporal scales in Section 3.2. The chapter concludes by summing up the material covered, and by re-iterating important findings of the experiments and the questions raised above (Section 3.3).

3.1 Importance of Multiple Scales

The first half of this section discusses the role of multiple scales in the spatial domain, which served as a foundation for the development of many successful feature detectors, to build motivation for the study of multiple temporal scales in visual tracking. The second half underscores open problems in the visual tracking domain that can be approached through learning and operating over multiple temporal scales.

Computer vision algorithms interpreting image data for applications like 3D reconstruction [Ladikos et al., 2008], [Furukawa and Ponce, 2009], pose estimation [Shakhnarovich et al., 2003],[Agarwal and Triggs, 2004],[Shotton et al., 2013], object recognition [Fergus et al., 2003], [Lazebnik et al., 2006], [Fei-Fei et al., 2007], and many more typically require feature detection in the early stages of processing. Feature detection is often considered vital because it is important to achieve invariance to geometric transformations such as translation, rotation, and scale and photometric transformations like illumination and exposure.

One of the early works in this direction was edge detection [Prewitt, 1970]. Edge detection might appear a relatively straightforward task, but it was observed that it is not simple to extract edge descriptors reliably. Typically, this was attributed to noise that can be reduced by smoothing the image before applying the edge detector [Canny, 1983],[Torre and Poggio, 1986]. Later, it was revealed that these problems actually stem from the fact that real-world objects contain different types of structures at different scales. To handle the multi-scale nature of real-world objects, multi-scale representations such as pyramids [Burt and Adelson, 1983], and scale-space [Witkin, 1984],[Koenderink, 1984],[Lindeberg, 1993] were proposed.

Perhaps the best-known work on multiple scales in the spatial domain is the scale-space theory proposed by Lindeberg [Lindeberg, 1994]. It states that the notion of scale is crucial when extracting semantically meaningful information and suppressing irrelevant variations from multidimensional signals. For instance, in terms of image data, the concept of scale is crucial when computing features and descriptors. Since real-world objects contain different types of structures at different scales, they may look different depending on the scale of observation. For example, the concept of a branch of a tree might be relevant at scales of a few centimeters to only a few meters, while the concept of a forest only make senses at the kilometer level. A robust vision system should be able to handle these scale variations.

When a vision system observes an unknown scene, there is no way to know in advance what scales will be useful for extracting meaningful information. A multi-scale representation of the image data, in which the original signal is embedded into a one parameter family of signals using scale as the parameter [Lindeberg, 2007], is therefore indispensable.

The hypothesis made here is that a similar approach could be adopted in visual tracking. In particular, a tracker operating over multiple temporal scales can generate a rich set of temporal information and allow complex variations (motion and appearance) to be captured. Problems like occlusion and abruptly varying target motion can be addressed with this formulation.

Occlusions can take place over different periods of time [Yilmaz et al., 2006]. A tracker operating at a single temporal scale (for instance, [Huttenlocher et al., 1993], [Koller-Meier and Ade, 2001],[Adam et al., 2006],[Ross et al., 2008],[Mei and Ling, 2009], and [Kwon and Lee, 2011]) cannot reliably handle the variability associated with periods of occlusion because it relies on a single temporal prior at the current time-point to estimate the target state. Here, a temporal scale is the duration of a specific sequence of moments in time e.g. $[t-1;t]$ or $[t-4;t]$, and the temporal prior associated with this is the information (estimated posterior or a motion prediction) propagated from the first member of the sequence to the last. This prior becomes unreliable if the duration of occlusion is larger than the temporal scale from which it is generated. Visual tracking over multiple temporal scales has the potential to solve this problem. Single scale trackers such as [Dellaert et al., 1999], [Khan et al., 2005], [Wu and Nevatia, 2006], [Ross et al., 2008], [Yang et al., 2009a], [Mei and Ling, 2009], [Kwon and Lee, 2010],[Babenko et al., 2011], [Jia et al., 2012], [Park et al., 2012], and [Xing et al., 2013] implicitly assume that occlusions are shorter than the temporal scale used. However, multiple scale trackers make the weaker assumption that any occlusion will be shorter than the longest temporal scale used [Yilmaz et al., 2006].

In unconstrained environments, target motion can exhibit complex variations [Yang et al., 2011]. In predictive tracking, learnt motion models describe the recent history of the target state - the most recent section of the target's path across the image plane. Trackers using, for example, a single linear motion model effectively represent the target path as a straight line, which obviously cannot capture and represent likely variation in this path. By building multiple motion models at multiple temporal scales, much as images are modelled at multiple spatial scales, it is possible to maintain a much richer description of the target path. The diverse set of models produced captures at least some of the complexity of that path and, when used to make predictions, the model set has the potential to represent variation in target motion better than any single model.

Appearance variations can also be learnt over multiple temporal scales [Li et al., 2008], [Xing et al., 2013]. A model generated from a shorter scale is more specific as it has only seen a few examples of target appearance, while a model learnt over a longer scale is more general since it has access to a wider range of examples of target appearance than the shorter scale model. While tracking, longer scale models can capture large appearance variations and so may avoid the drift problem better than shorter scale models, but their likelihood response is not smooth and is costly to learn. Shorter scale models may catch smooth appearance variations better than longer scale models, and their likelihood response is relatively smooth and quicker to learn. Therefore, a set of appearance models, each learnt over a different temporal scale, should be able to handle variability in appearance better than any single model.

The following section tests the hypothesis, both in terms of motion and appearance, that a set of models, learnt over different temporal scales, can provide better prediction performance than any single model.

3.2 Probing motion and appearance variations in the ground truth data

It is hypothesized that when utilizing multiple temporal scales, there are often better models available than the model based on the most recent, minimal duration state history. The model built at the minimal duration state history is taken as the basis of comparison because it is common practice in predictive tracking to use this model, especially when describing target motion.

To test the aforementioned hypothesis, experiments were performed evaluating the ability of each of a set of models, learned over state histories of different lengths, to predict

motion and appearance in the ground truth data associated with seventeen challenging video sequences. These ground truth experiments demonstrate the feasibility of obtaining improved performance by learning over multiple temporal scales.

Of seventeen image sequences, fourteen are publicly available (*Car*, *Basketball*, *Deer*, *Lemming*, *Shaking*, *Singer1*, *Iron*, *Girl*, *Trellis*, *Football*, and *Mountainbike* are from [Wu et al., 2013], *Panda* is from [Kalal et al., 2012], and *Singer1(low frame rate)*, and *Skating1(low frame rate)* are from [Kwon and Lee, 2010]) and three are our own (*Squash*, *toy2* and *Ball1*). These video sequences contain different targets, such as the head of a vocalist, a toy tied to a thread, a lady skating, an animal sprinting, etc. Since the videos were shot in unconstrained environments, the targets exhibit complex motions either due to their own movement, camera motion, or low frame rate and miscellaneous changes in appearance caused by factors such as out-of-plane rotations and illumination variations.

Ten sequences were used to evaluate motion prediction, and seven to assess appearance prediction. Table 3.1(a), and Table 3.1(b) highlight the challenging aspects of the videos associated with motion prediction and appearance prediction, respectively.

TABLE 3.1: **Challenging aspects of videos used for evaluating motion and appearance prediction.**

(A) Videos used for evaluating motion prediction.

Sequence	Main Challenges
toy2	Target accelerates, decelerates, and suddenly changes motion direction
Deer	Abruptly varying target motion due to target itself
Lemming	Target accelerates and decelerates
Shaking	Smooth target motion
Car	Abrupt target motion due to low frame rate and camera motion
Singer1(low frame rate)	Abrupt target motion due to low frame rate
Skating1(low frame rate)	Abrupt target motion due to low frame rate
Ball1	Abruptly varying target motion due to target itself
Squash	Irregular target motion due to target itself
Mountainbike	Smooth target motion

(B) Videos used for assessing appearance prediction.

Sequence	Main Challenges
Singer1	Illumination Variation, and Out-of-Plane Rotation
Iron	Illumination Variation, Out-of-Plane Rotation, Out-of-View, and Partial Occlusion
Girl	Illumination Variation, Out-of-Plane Rotation, and Partial Occlusion
Trellis	Illumination Variation, and Out-of-Plane Rotation
Football	Out-of-Plane Rotation, and Partial Occlusion
Panda	Illumination Variation, Out-of-Plane Rotation, Partial and Full Occlusion
Football1	Out-of-Plane Rotation

3.2.1 Learning Motion Over Multiple Temporal Scales

This section evaluates the position(location) prediction performance of a set of motion models learned over different temporal scales. To examine location prediction, multiple linear motion models are learned from the recent history of target's state over different model-scales, and then these learned models are used to predict target state at the next time-step. The model-scale is the duration of a sequence of recently estimated target states (See Fig. 3.1).

A linear motion model M is learned at a given model-scale m separately for the x-dimension, and y-dimension of the target's state. The x, and y part of the target state are considered uncorrelated. This is done so that the basic idea of the proposed approach can be evaluated properly. They may be correlated under some circumstances, for example, when a target's path across image plane is roughly diagonal. Taking correlations among state's component into account while learning models could be an avenue for future work.

Let $Z^m = \{\hat{x_n}\}_{n=t-m+1}^{n=t}$ represent a sequence of recently estimated x-locations of target states at model-scale m. Similarly, let $W^m = \{\hat{y_n}\}_{n=t-m+1}^{n=t}$ denote a sequence of recently estimated y-locations of target states at model-scale m. For this preliminary study, Z^m and W^m are obtained by manual annotation. Given M learned at model-scale m at time t, for x-location of target state it is written as:

$$\tilde{x}_n = \phi_{\hat{x}_t}^m + \tau_{\hat{x}_t}^m n, \tag{3.1}$$

where τ is the slope, ϕ the intercept, and $\hat{x}_t$ denotes that the model parameters have been learnt using a sequence of recently estimated x-components of target states whose last member is $\hat{x}_t$. Model parameters can be learned inexpensively via ordinary least squares (OLS) applied to Z^m. Along similar lines, given M learned over model-scale m at time t, for y-location of target state it is written as:

$$\tilde{y}_n = \phi_{\hat{y}_t}^m + \tau_{\hat{y}_t}^m n, \tag{3.2}$$

where the parameters τ, ϕ, and $\hat{y}_t$ in this equation has the same interpretation as described for Eq.3.1 and model parameters are learned via OLS applied to W^m.

A linear motion model is preferred over higher-order models in this study for several reasons. Since it is possible to learn a linear function on as few as two data-points, small model-scales can be included in the model set. With a linear motion model, the risk of over-fitting is largely avoided in our case, as the largest model-scale in the model set is on the shorter side. A relatively large number of data points are required for learning

a higher order model. Although extrapolation is not considered safe for interpolating polynomial functions, it is usually more reliable with a linear motion model than higher order models [Clymo, 2014].

Here, we denote a set of motion models corresponding to model-scales 1, 2, 3, 4 and 5 at time t by $\mathbf{M}_t = \{M_t^1, M_t^2, M_t^3, M_t^4, M_t^5\}$. Fig. 3.1 illustrates 5 motion models constituted over model-scales 1, 2, 3, 4 and 5. Motion models corresponding to model-scales 2, 3, 4, and 5 are learned using a linear function. As there is only a single previous state estimate for model-scale 1 (corresponding to the first-order Markov Chain assumption), a linear function cannot be learned. Therefore, the motion model belonging to model-scale 1 predicts the next location by simply adding Gaussian noise to the previous location. Although these model-scales were chosen empirically, they cover the range needed to represent (while predicting) the variations in the target's path, which form patterns of different lengths.

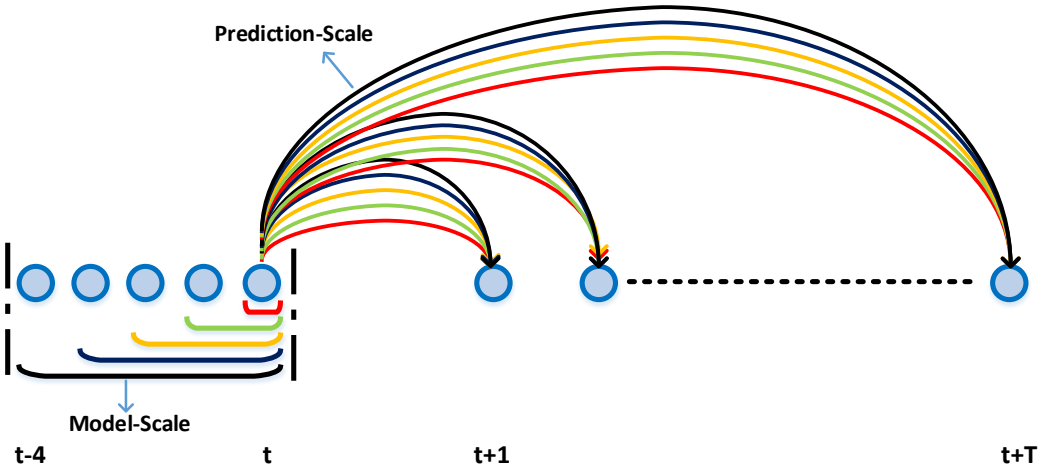


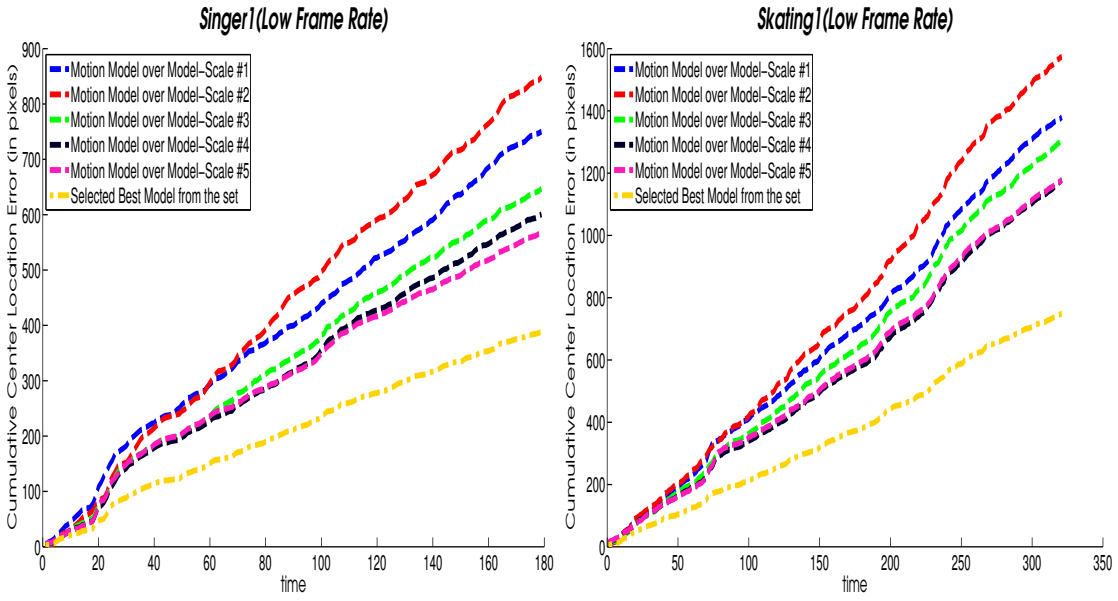
FIGURE 3.1: **Learning and Predicting Motion Over Multiple Temporal Scales.** Multiple motion models are learned from the recent history of estimated states at different temporal scales, and each model is applied over multiple temporal scales in the future. In this figure, 5 motion models corresponding to 5 different temporal scales are shown at time t, where 1 out of 5 is a standard first-order Markov model, and 4 others are learned on 4 different sequences comprising recently estimated states of lengths 2, 3, 4 and 5, respectively, and each model predicts target state T time-steps ahead.

Each member of the set $\mathbf{M}_t$ can be used to predict target location multiple time-steps ahead in the future e.g. $[t+1, t+6]$. Here, however, learned models are used to generate predictions only at time $t+1$ to test the hypothesis stated in the beginning of this chapter i.e. learning motion over different model-scales can add value in terms of improvement in position prediction performance. For each predicted location, the error is computed by finding its Euclidean distance from the ground truth location at time

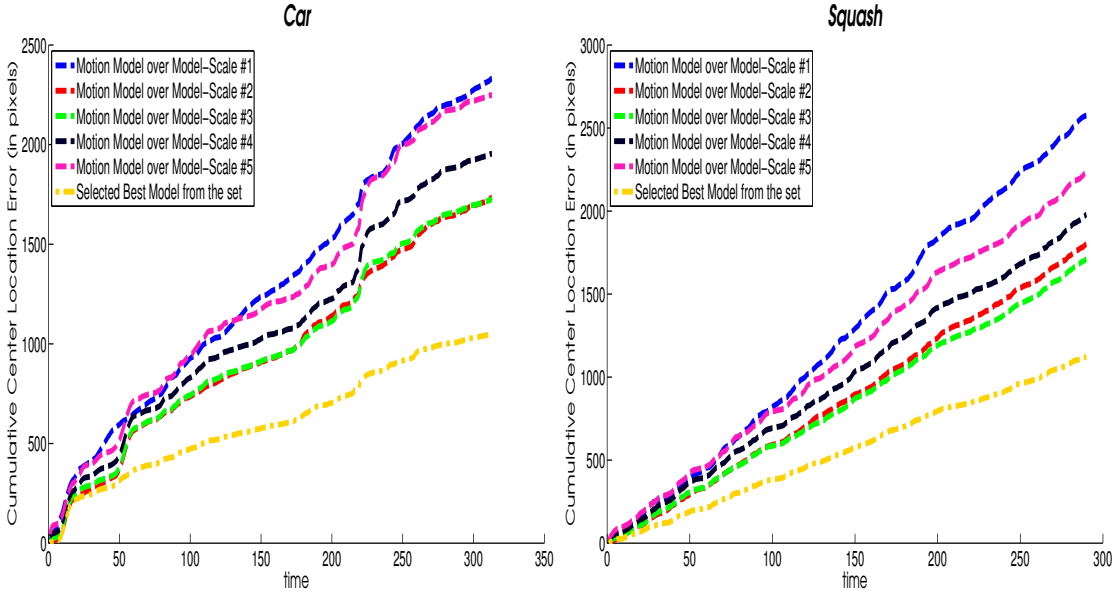
$t+1$. Cumulative centre location error (over time) is used to assess the performance since it is easier to interpret than raw centre location error (over time). The former summarizes the performance (through accumulation) at each time-point and thus, makes comparison among competing methods more interpretable than the latter. Fig. 3.2 plots the cumulative location error associated with each individual motion model belonging to set $\mathbf{M}_t$ and that obtained by selecting the best performing model from this set for each time-point. Note that this best model is chosen using oracle-like knowledge of the ground truth, and selecting the right motion model is a major challenge.

The results demonstrate that the position prediction performance improves when the best model is selected from a set of models learnt over multiple scales, as compared to any fixed, individual model. Thus, if one knows which model to pick, access to a set of models learned at multiple model-scales can significantly improve performance. It is interesting to note that with this access the performance gain is not only substantial for sequences containing abrupt motion but is also encouraging for the sequences in which the target motion varies relatively smoothly. In addition, the results reveal that for most of the sequences considered, the motion models corresponding to different temporal scales maintain their performance ranking over time, although that ranking is different for each video class, and thus context-dependent. For example, the ranking of individual models is fixed across Fig 3.2(a), which displays results for low frame rate videos, but it is different in Fig. 3.2(b), which plots results under abrupt motion variations. In a video class, large variations in the target motion are produced by the same factor(s) such as camera movement and low frame rate.

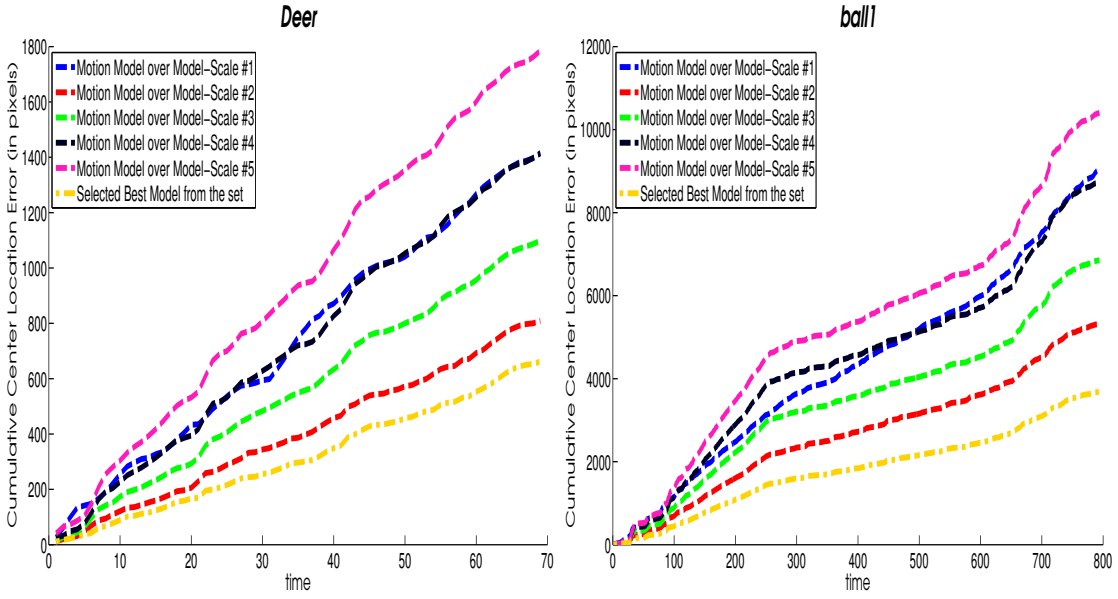
Fig. 3.2(a) shows motion prediction performance given low frame rate videos. Perhaps the most noteworthy point is that in both videos the ranking of individual models is almost the same, and upon selecting the best model from the model set the performance gain compared to the best individual model is almost the same. On an individual basis, the model corresponding to the largest model-scale performed the best, and as the model-scale decreases, the accuracy of the associated model decreases. This might be due to the fact that when a video is captured at a low frame rate, or downsampled, most of the time the resulting variation in motion remains smooth. More often than not, longer scale models are a better fit to periodic variations in motion. As an example, Fig. 3.3(a) displays the temporal evolution of target state in the first 100 frames of the *skating1(lfr)* sequence, which contains (nearly)smooth motion variations, and Fig. 3.3(b) plots prediction performance of motion models corresponding to model-scales 1, 2, 3, 4, and 5 over these frames. It can be seen that, generally, longer scale models have better prediction performance than shorter scale models.



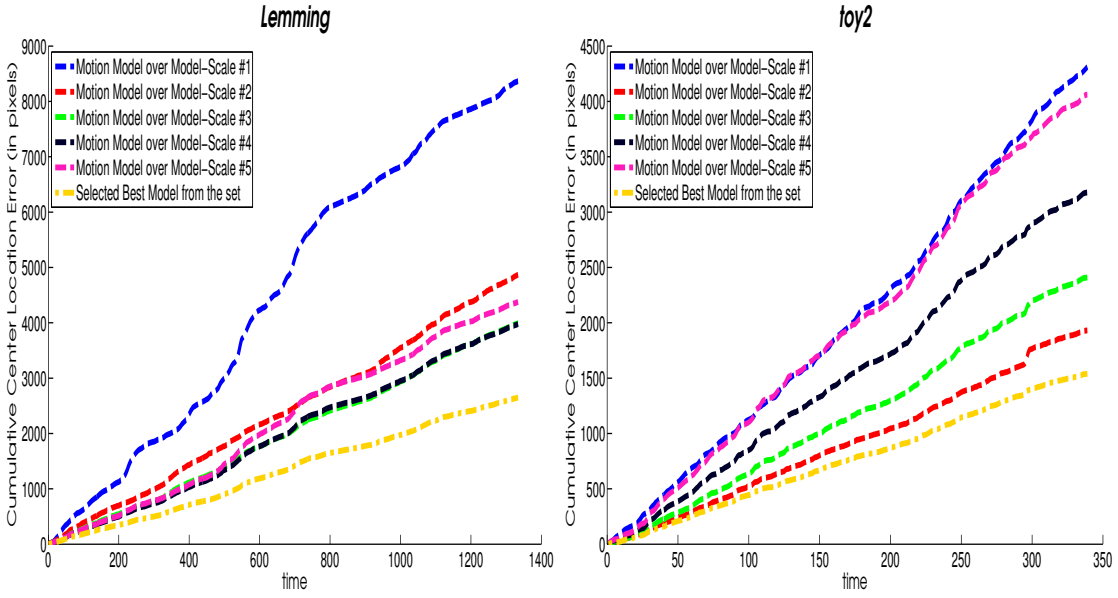
(A) Prediction performance when there are abrupt motion variations due to low frame rate (∼1/3 of the original frame rate of the sequence.



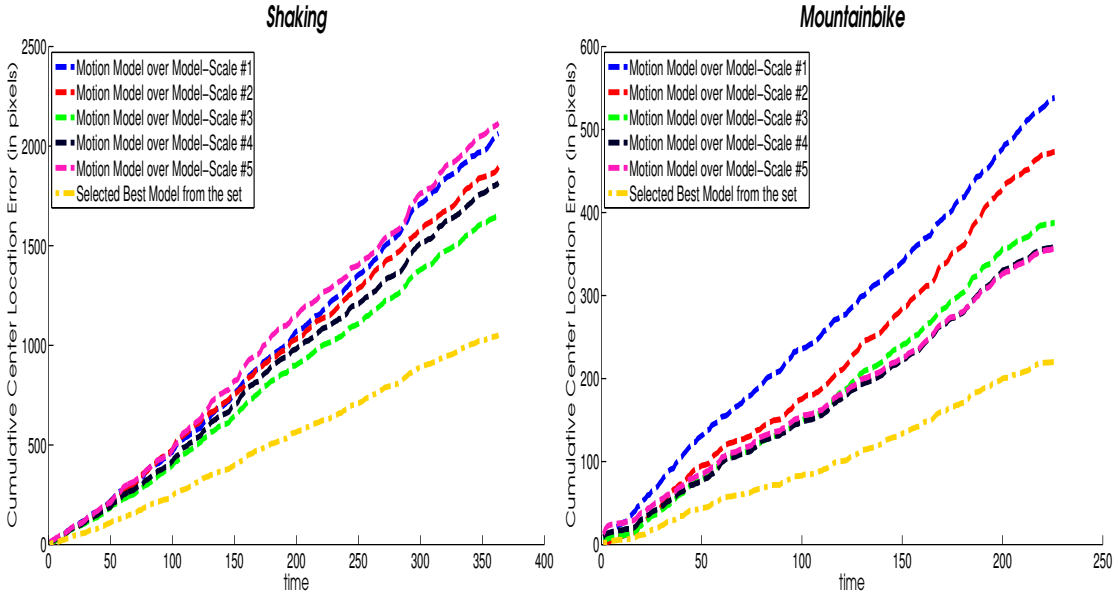
(B) Prediction performance when there are abrupt motion variations due to camera motion, or the target itself.



(C) Prediction performance when there is abrupt motion due to target itself.



(D) Prediction performance when the target accelerates, decelerates, or suddenly changes direction.

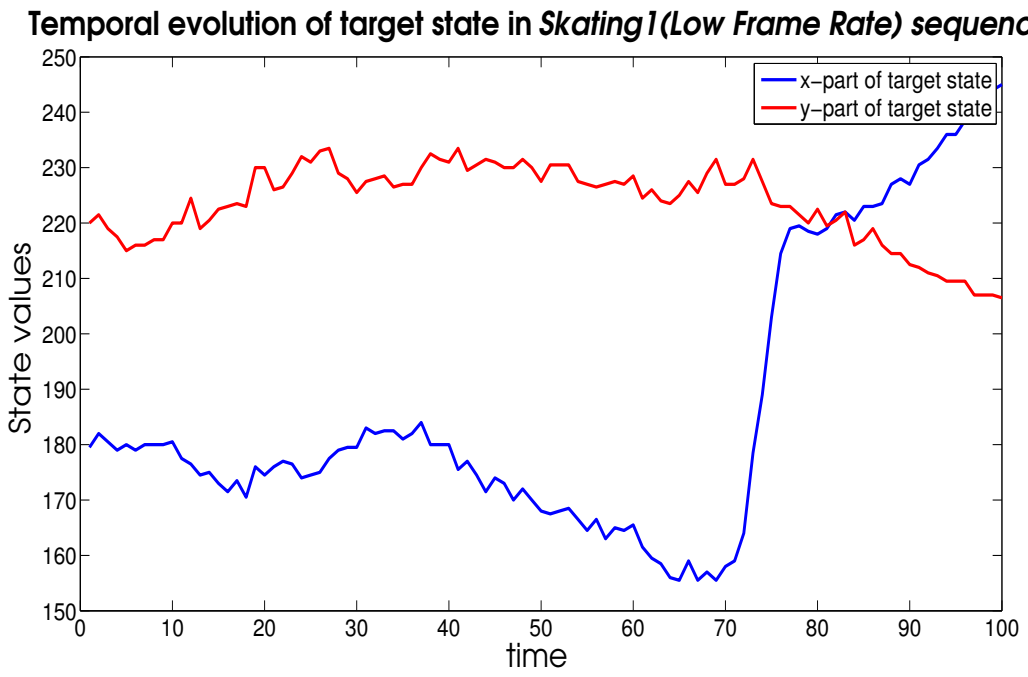


(E) Prediction performance in case of smooth motion variations.

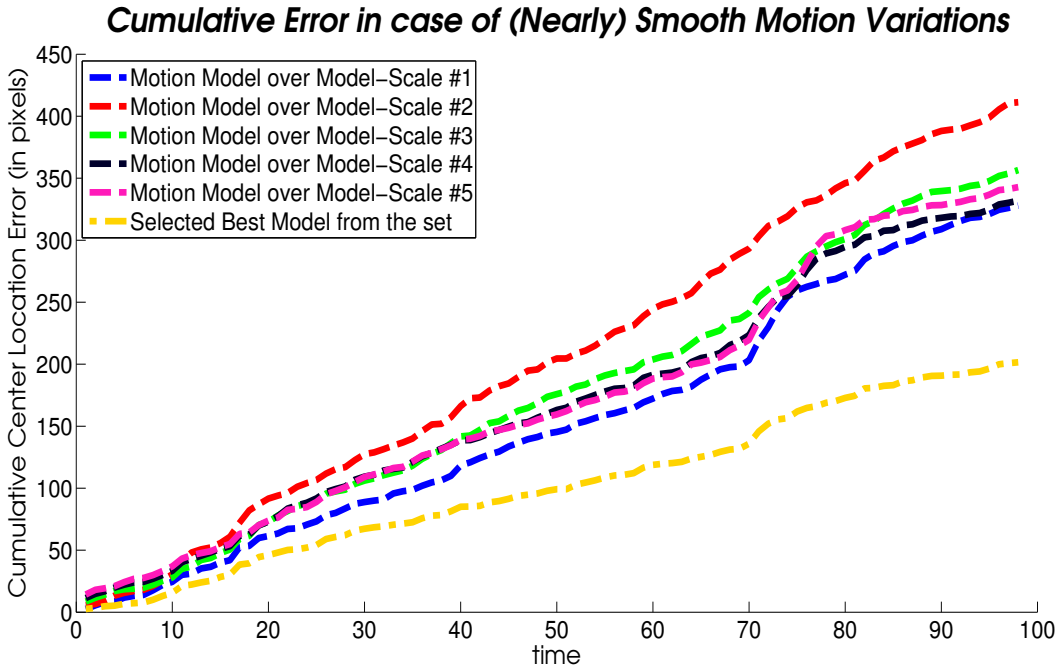
FIGURE 3.2: **Performance comparison in case of motion prediction on ten challenging video sequences.** Each plot illustrates the cumulative center location error over time for the selected best motion model from a set and the individual motion models.

Fig. 3.2(b) and Fig. 3.2(c) plot location prediction performance when there are abrupt motion variations due to camera shake and/or the target itself. Interestingly, again, for the videos included in each figure, individual models keep the same order in terms of performance. However, for videos in this class, the shorter scale models, except the shortest, outperform the longer scale models. The same can be observed in situations where motion variations are due to acceleration, deceleration, and sudden changes in the direction of motion. For instance, in the right panel of Fig. 3.2(d), which plots results obtained from the *toy1* sequence, it can be seen that when switching from the longer scale models to the shorter scale models prediction performance improves. It could be argued that in these sequences, the motion of the target is quite erratic; typically, it does not make a smooth pattern over time. As an example, Fig. 3.4(a) displays temporal evolution of target state in the first 100 frames of the *ball1* sequence, which contains unpredictable motion variations, and Fig. 3.4(b) plots performance of motion models corresponding to model-scales 1, 2, 3, 4, and 5 over these frames. It can be seen that, generally, shorter scale models outperform their longer scale counterparts.

An improvement of 300 pixels and 128 pixels cumulative location error was observed in the *Shaking* and *Mountainbike* sequences, which contain smooth motion variations, when selecting the best-performing from a set of models, derived from multiple model-scales, in comparison to a single member of this set (See Fig. 3.2(e)). The difference in

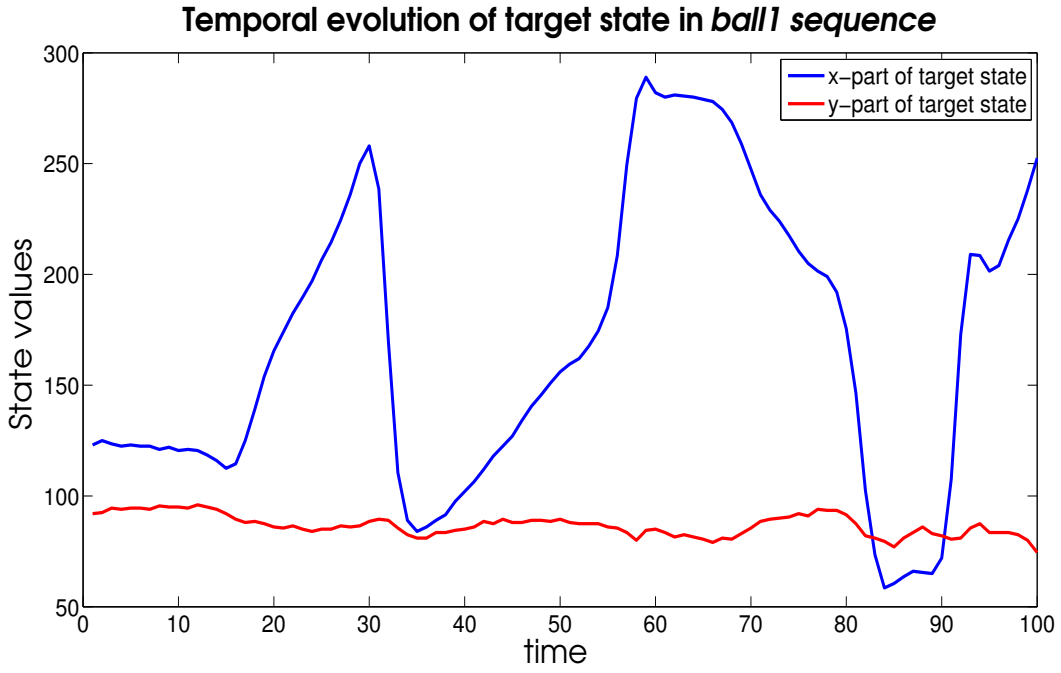


(A) Temporal Evolution of target state in the first 100 frames of the skating1(low frame rate) sequence.

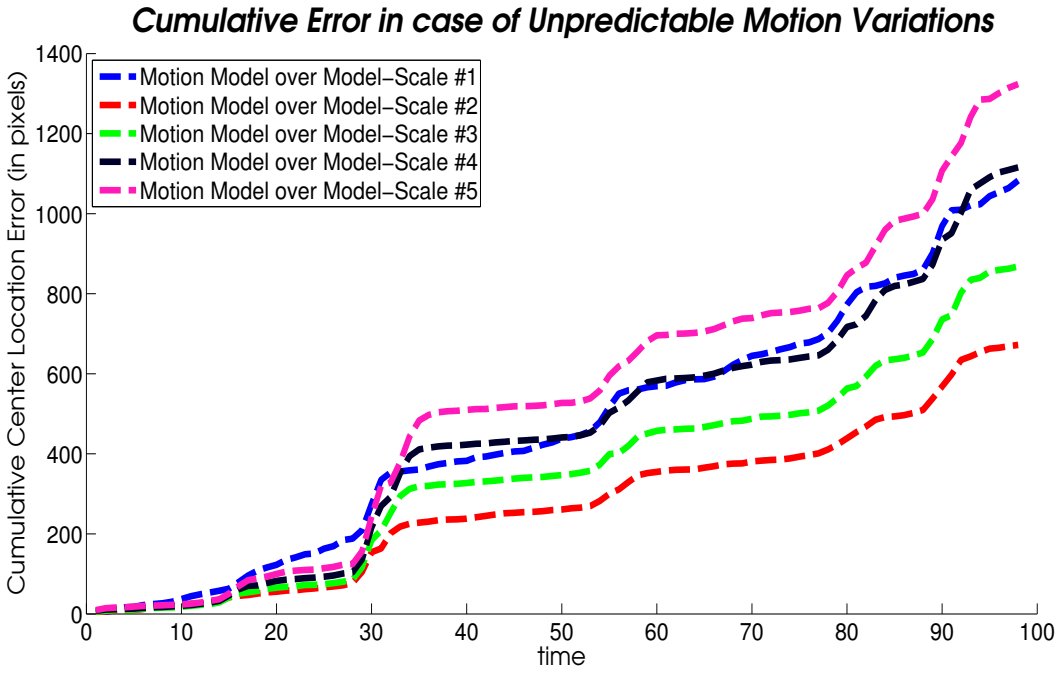


(B) Performance of models corresponding to model-scales 1,2,3,4, and 5 in these frames.

FIGURE 3.3: **Position prediction performance under (nearly)smooth motion variations.**



(A) Temporal Evolution of target state in the first 100 frames of *ball1* sequence.



(B) Performance of models corresponding to model-scales 1,2,3,4, and 5 in these frames.

FIGURE 3.4: **Position prediction performance under unpredictable motion variations.**

individual performances of the models is not sizable, especially in the *Shaking* sequence, but longer scale models still surpass their shorter scale fellows.

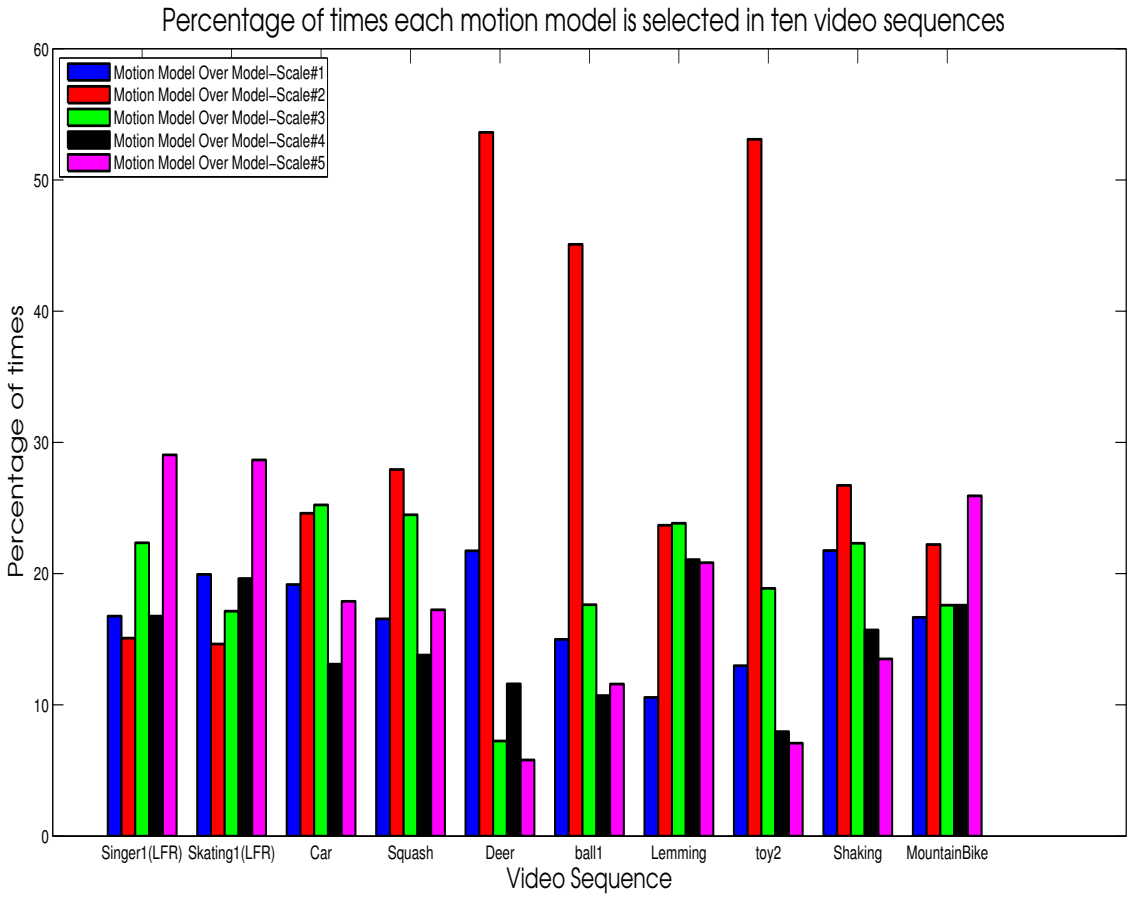


FIGURE 3.5: **Selection percentage for each motion model in ten video sequences.** The figure illustrates percentage of times each motion model is selected in ten video sequences.

The cumulative error associated with a motion model over the course of a video sequence does not always reflect its selection percentage i.e. the percentage of times that motion model makes the most accurate prediction at a given time-step in a video sequence. A few very inaccurate predictions may give motion model a high cumulative centre location error, but, due to it making slightly better predictions than the others most of the time, this model might have a relatively high selection percentage. Fig. 3.5 shows the percentage of times each motion model is selected in ten video sequences. Improved performance of longer scale models over shorter scale models in videos where motion variations usually form smooth patterns similar in length to the longer scales used can be claimed, but in a loose sense. In contrast, the dominance in terms of performance of shorter scale models over longer scale models in videos where motion variations are mainly unpredictable can be claimed much more confidently. For instance, in the *Deer* sequence, models derived from model-scale 2 and 1 made the most accurate prediction

52% and 22% of times, respectively, whereas models constituted at model-scale 5 and 4 delivered the most accurate prediction 8% and 12% of times, respectively.

3.2.2 Learning Appearance Over Multiple Temporal Scales

Like motion, target appearance is subject to variations over time in unconstrained tracking environments. For an appearance model to stay valid for extended periods of time, it is imperative to capture these variations. Having seen the advantages of access to a set of motion models, learned over different model-scales, over any individual model, this section will apply the same concept, learning over multiple temporal scales, to appearance modelling, and examine the effect on performance. Appearance models are learned over different model-scales from the appearances corresponding to recently estimated target states. These learned models are then used to predict target appearance at the next time-step.

It is important to note the difference between the updated appearance model at time t after estimating the state, which is usually found in the tracking literature, and the predicted appearance at time $t+1$, which will be introduced in this section. Tracking approaches [Han and Davis, 2005][Grabner and Bischof, 2006],[Ross et al., 2008], [Babenko et al., 2009],[Mei and Ling, 2009],[Kwon and Lee, 2010], that update model at time t assume that the true target appearance at time $t+1$ will not be very different from this updated model, whereas the approach that will be described in this section relaxes this assumption and hallucinates an appearance at time $t+1$ that might be substantially different compared to the evidence-based model of the appearance at time t.

A colour histogram [Pérez et al., 2002] was used to encode target appearance in these experiments. It has three desirable features in terms of tracking. Firstly, it is computationally cheap to compute, and is particularly suitable for search mechanisms that search a large state space. Secondly, the resulting feature vector is quite compact, which is usually desirable for learning methods. Finally, it is robust to modest appearance variations and has proven useful in tracking a variety of different targets [Pérez et al., 2002],[Kristan et al., 2010].

Formally, the colour histogram is used to encode a target's appearance in the rectangular region $R(q_t)$ specified by the state q_t. The state at time t is given by $q_t = \{x_t, y_t, s_t\}$, where x_t, y_t, and s_t denote the x, y location, and scale of the rectangular region, respectively. In terms of a feature vector, the colour histogram can be represented as $\mathbf{h}(q_t) = (h^u(q_t))_{u=1\ldots N}$, where N is the total number of bins in a colour histogram.

To model the target appearance at the model-scale m, the following methodology is employed. Let $H^m = \{\mathbf{h}(\hat{q_n})\}_{n=t-m+1}^{n=t}$ be a sequence of length m comprised of colour histograms computed at a sequence of recently estimated target states $\{\hat{q_n}\}_{n=t-m+1}^{n=t}$. Similarly, let $b_u^m = \{(h^u(\hat{q_n})\}_{n=t-m+1}^{n=t}$ represents a sequence of m histogram bin counts corresponding to the u^{th} bin of the colour histogram $\mathbf{h}$. The top row of Fig. 3.6 graphically illustrates make-up of H^m, and b_u^m. For each b_u^m, a linear function as described in Eq. 3.1 is learned. A set of learned linear functions on b_u^m, where $u = 1...N$, defines a learned appearance model A_t^m at time t over temporal scale m. In other words, A_t^m comprises N learned linear functions over N sequences each of length m, where each sequence consists of m histogram bin counts corresponding to the u^{th} bin of the colour histogram $\mathbf{h}$. The bottom row of Fig. 3.6 graphically demonstrates the composition of A_t^m.

To evaluate the primary idea of learning appearance over multiple temporal scales and for the sake of simplicity, the features or the bins of a histogram are considered independent. We acknowledge that this is a strong assumption. It is quite likely for the features to be correlated in some way, and taking this into account while learning could yield improved models.

Here, a set of appearance models corresponding to model-scales 1, 5, and 9 at time t is symbolized by $\mathbf{A}_t = \{A_t^1, A_t^5, A_t^9\}$. Appearance models associated with scales 5, and 9 are learned using the aforementioned methodology. The model at scale 1 utilizes the previously estimated appearance to be the predicted appearance at the next moment in time.

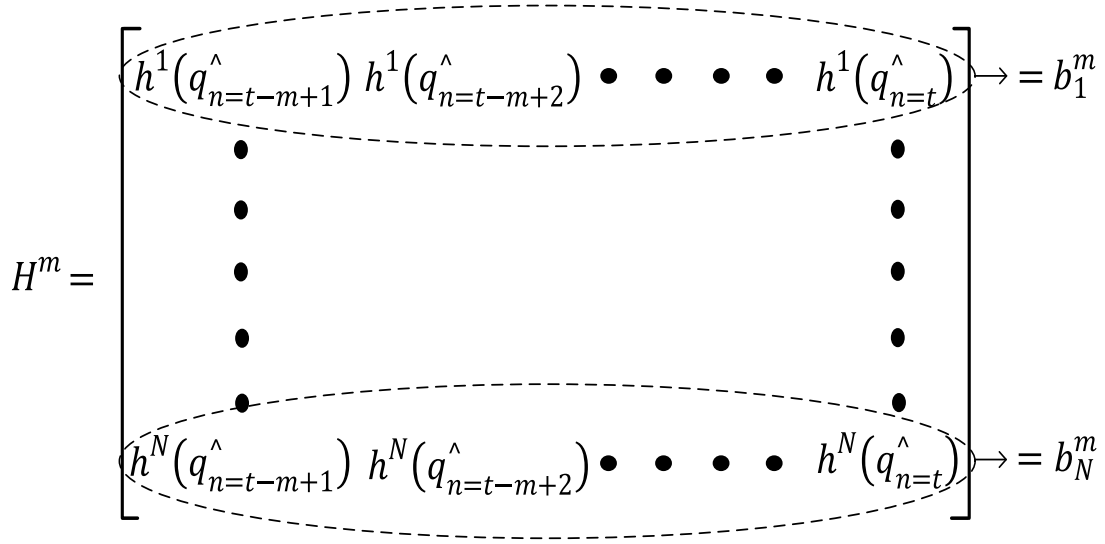
These model-scales were chosen as existing tracking methods based on adaptive appearance models either use the immediate previous observation from time-point $t-1$ [Grabner and Bischof, 2006, Han and Davis, 2005, Matthews et al., 2004] or observations from some sequence of past time-points e. g. $[t-1; t-5]$ [Kwon and Lee, 2010, 2011, Ross et al., 2008] for modelling appearance.

Every member of the set $\mathbf{A}_t$ predicts a colour histogram (target appearance) $\tilde{\mathbf{h}}_{t+1}$ at time $t+1$. For each predicted $\tilde{\mathbf{h}}_{t+1}$, the squared Bhattacharyya distance [Comaniciu et al., 2003] to the colour histogram extracted at the ground truth state $\mathbf{h}(g_{t+1})$ is computed at time $t+1$:

$$D_{bhat}^2(\tilde{\mathbf{h}}_{t+1}, \mathbf{h}(g_{t+1})) = 1 - \sum_{u=1}^{N} \sqrt{\tilde{h}_{t+1}^u h^u(g_{t+1})}, \tag{3.3}$$

where g_{t+1} indicates the ground truth state at time $t+1$.

Fig. 3.7 plots the cumulative Bhattacharyya distance over time of the individual appearance models belonging to the set $\mathbf{A}$ and the best performing model from this set



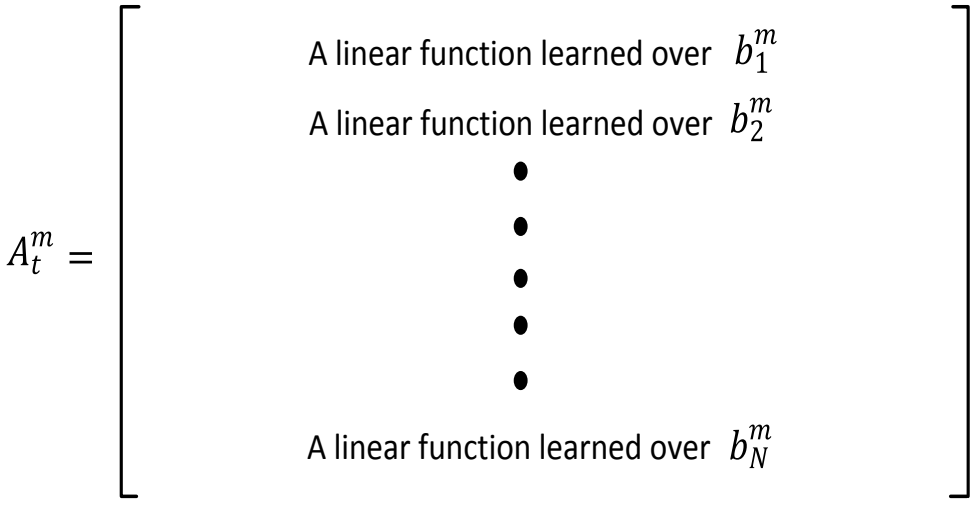


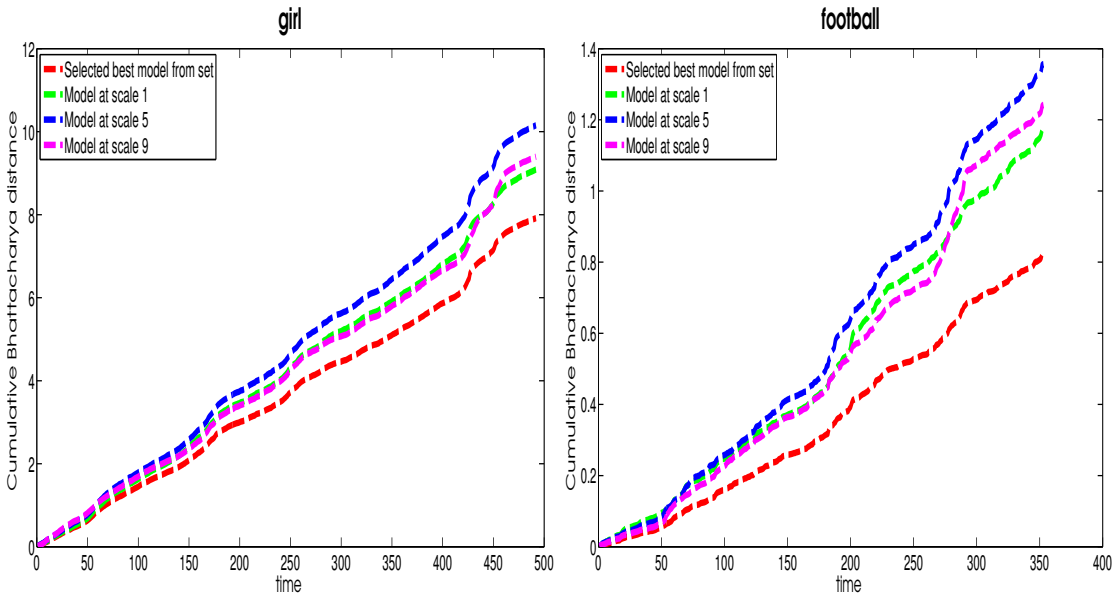
FIGURE 3.6: **Appearance model over model-scale** m**.** The figure graphically illustrates the composition of appearance model over model-scale m.

from seven video sequences. The model with the lowest distance at a given time is taken as the best model at that time.

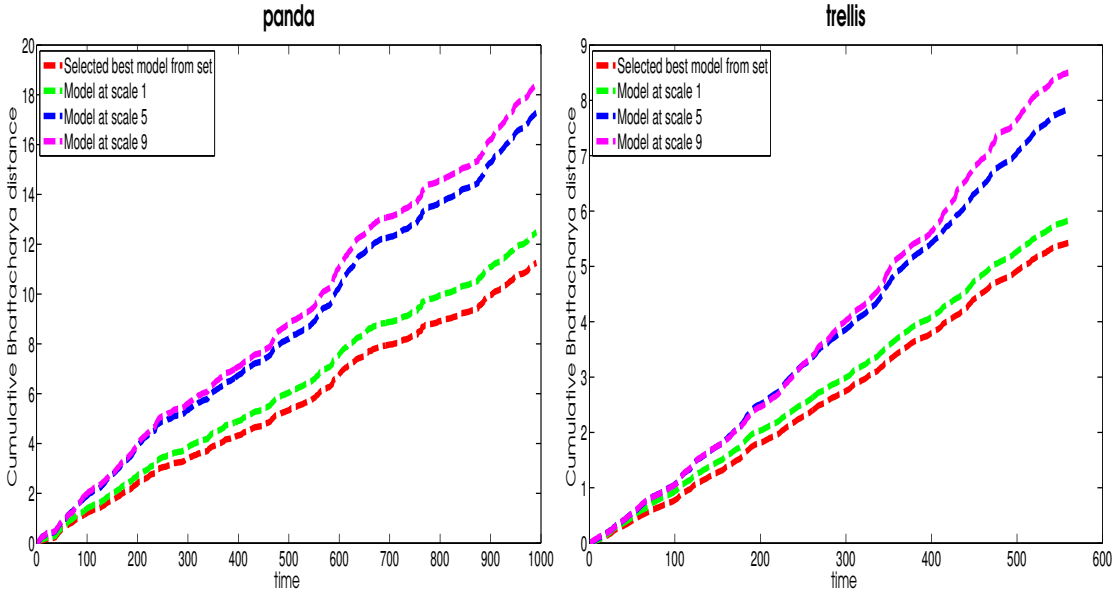
Results reveal that better appearance predictions can be made if a set of appearance models learned at different temporal scales is used in comparison to an individual, fixed appearance model. Again, this relies on being able to select the right model from **A**. In addition, the results demonstrate that unlike motion models, the appearance models corresponding to different model-scales change their ranking over time in four out of seven sequences.

Appearance models generated from multiple model-scales improve appearance prediction performance in comparison to any individual model, but this performance gain is low compared to the gain made when this concept was applied to motion models. The dimensionality of the feature space (110D) when learning appearance is very high in comparison to the dimemsionality of state space (2D) when learning motion. However, the learning method used both for motion and appearance models is the same, and does not consider correlation among variables. This assumption affects appearance modelling more than the motion modelling, due to high dimensionality of the feature space. As a result, the learned appearance models produce appearances that are not as accurate as the states predicted by the learned motion models.

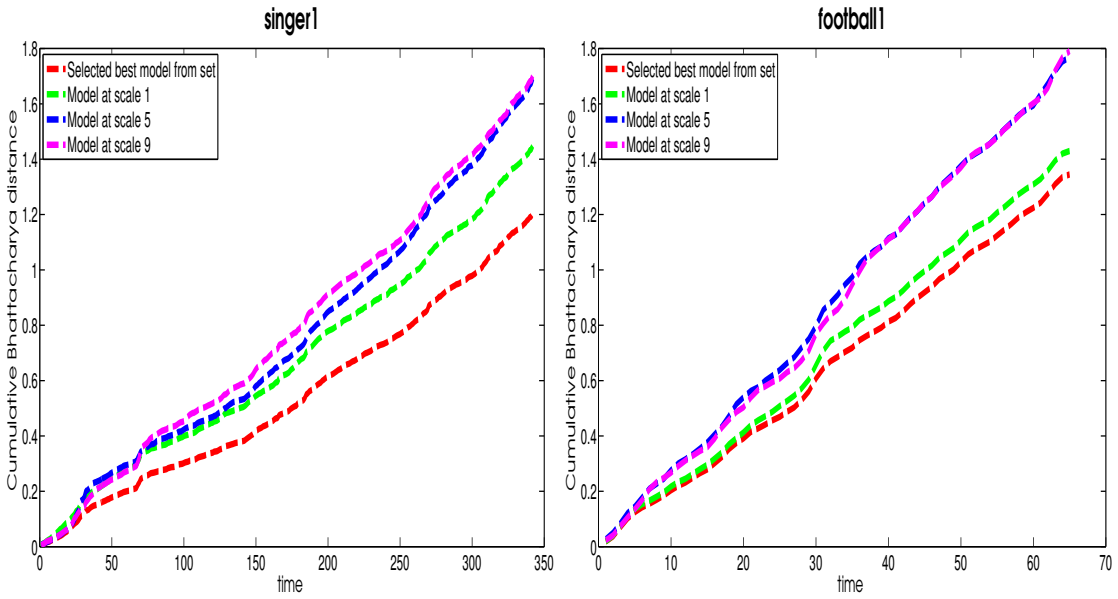
Fig. 3.7(a) shows appearance prediction performance in the *girl*, and *football* sequences. The ranks of appearance models corresponding to different model-scales are quite similar in both the sequences, but the gain in performance upon using the model set is apparently higher in the *football* sequence than the *girl* sequence. In both the sequences, the appearance model corresponding to model-scale 9 show slightly better performance than the model over model-scale 1 for more than two-third of the total duration, after which they switch ranks and continue unchanged till the end. The aforementioned trend is more evident in the *football* sequence. In contrast, the appearance model built over model-scale 5 does not change its rank throughout either sequences. A somewhat improved performance of the appearance model over model-scale 9 compared to its counterpart over model-scale 5 might be due to the fact that the variations in most of the variables (bin values) of the histogram form regular patterns of different time duration, which suit long scale models well. As an example, the top row of Fig. 3.8(a) shows the temporal evolution of bin values in the the first 200 frames of the *girl* sequence, and the bottom row of the Fig. 3.8(a) plots cumulative Bhattacharyya distance for appearance models corresponding to model-scales 5, and 9. Although there is not much difference in the performance of two models initially, model over scale 9 has somewhat better performance than the model over scale 5 for the most part.



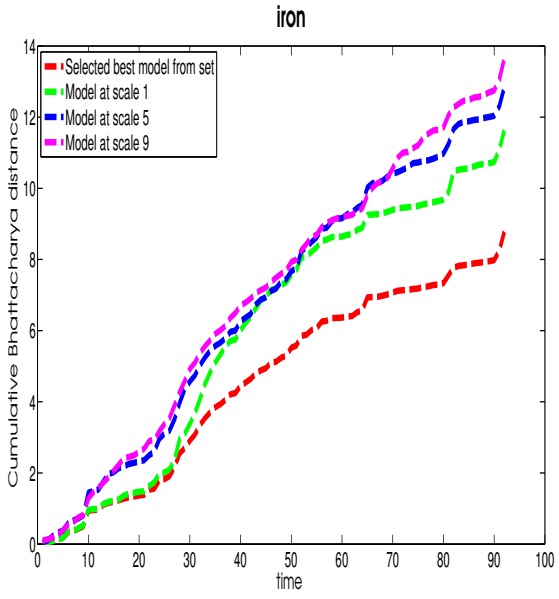
(A) Prediction performance when there are illumination variations, out-of-plane rotations, and partial occlusions.



(B) Prediction performance when there are illumination variations, out-of-plane rotations, and partial and full occlusions.

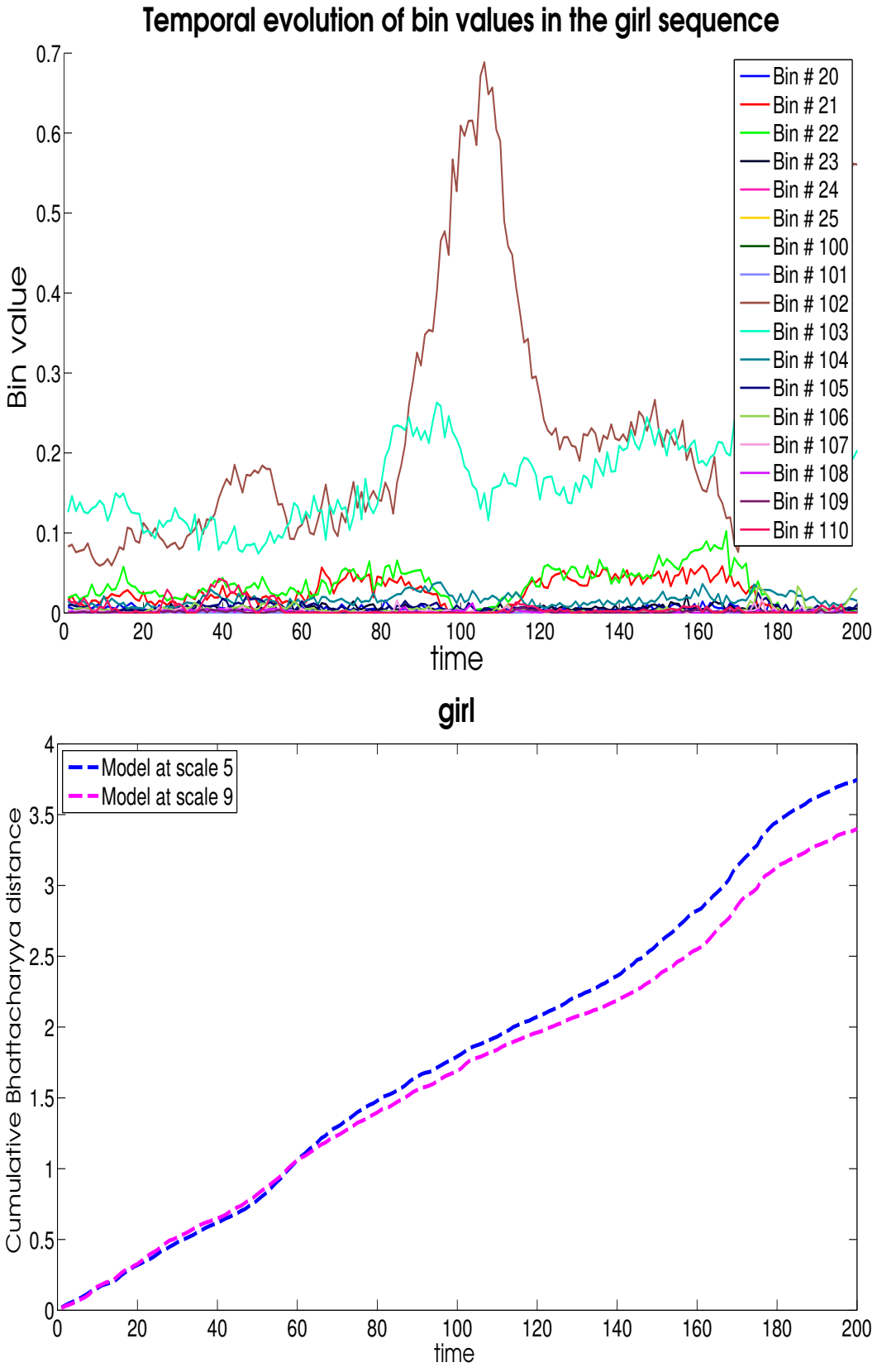


(C) Prediction performance in case of illumination variations and out-of-plane rotations.

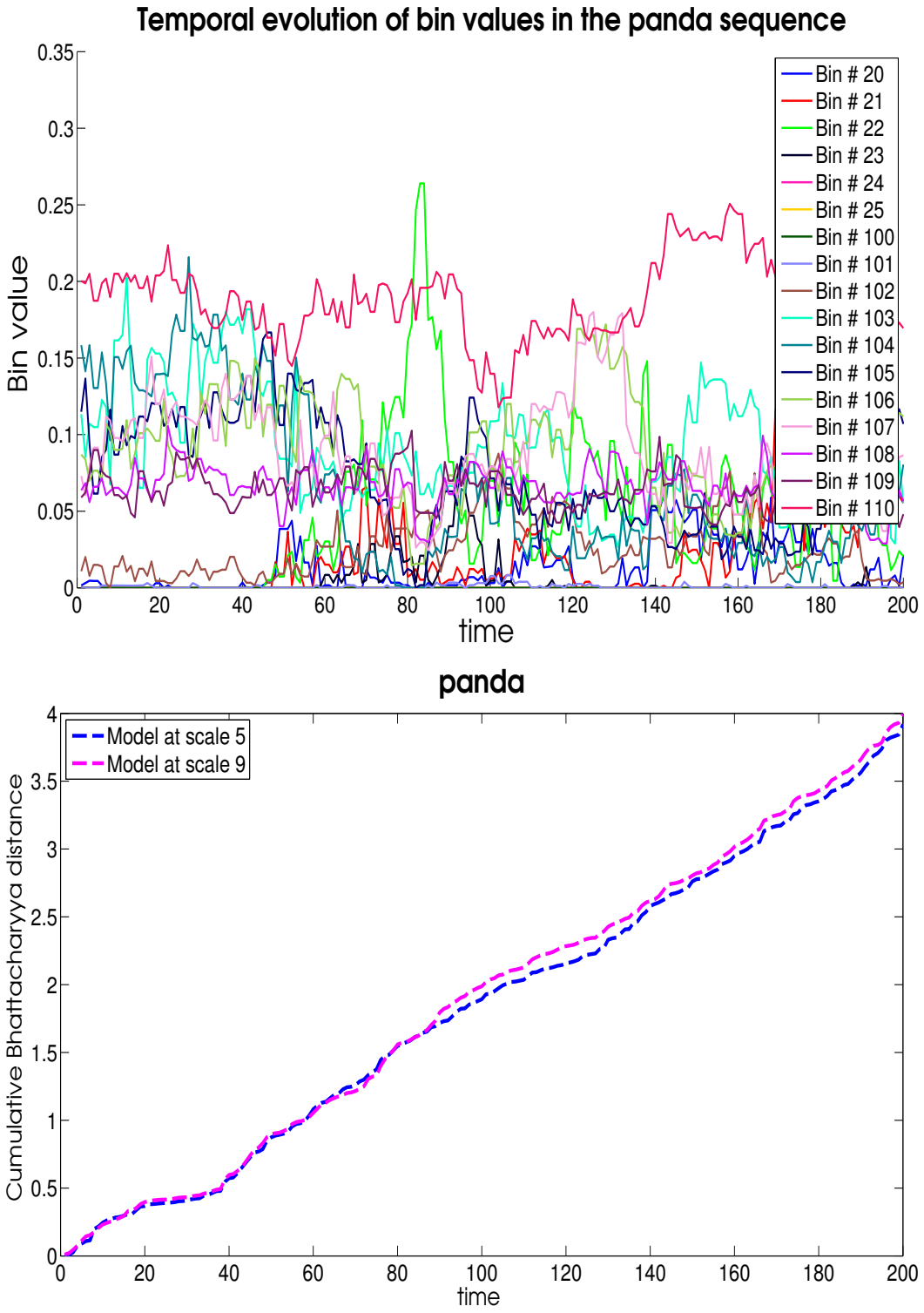


(D) Prediction performance in case of out-of-plane rotations, partial occlusions, illumination changes, and when some portion of the target leaves the view.

FIGURE 3.7: **Performance comparison in case of appearance prediction on seven challenging video sequences.** Each plot illustrates the cumulative Bhattacharyya distance over time for the selected best appearance model from a set and the individual appearance models.



(A) Temporal evolution of bin values in the first 200 frames of the *girl* sequence (top), and the performance of appearance models corresponding to model-scales 5, and 9 in these frames (bottom).



(B) Temporal evolution of bin values in the first 200 frames of the *panda* sequence (top), and the performance of appearance models corresponding to model-scales 5, and 9 in these frames (bottom).

FIGURE 3.8: **Performance comparison between appearance models derived from model-scale 5, and 9 under two different types of variations in bin values.** The plots in Fig. 3.8(a) show that when the variations in bin values in some way form regular patterns of different duration, then the appearance model over model-scale 9 performs slightly better than the appearance model over model-scale 5, while the plots in Fig. 3.8(b) reveal that under almost unpredictable (irregular) variations of the bin values, the appearance model over model-scale 5 performs a little superior to the appearance model corresponding to model-scale 9.

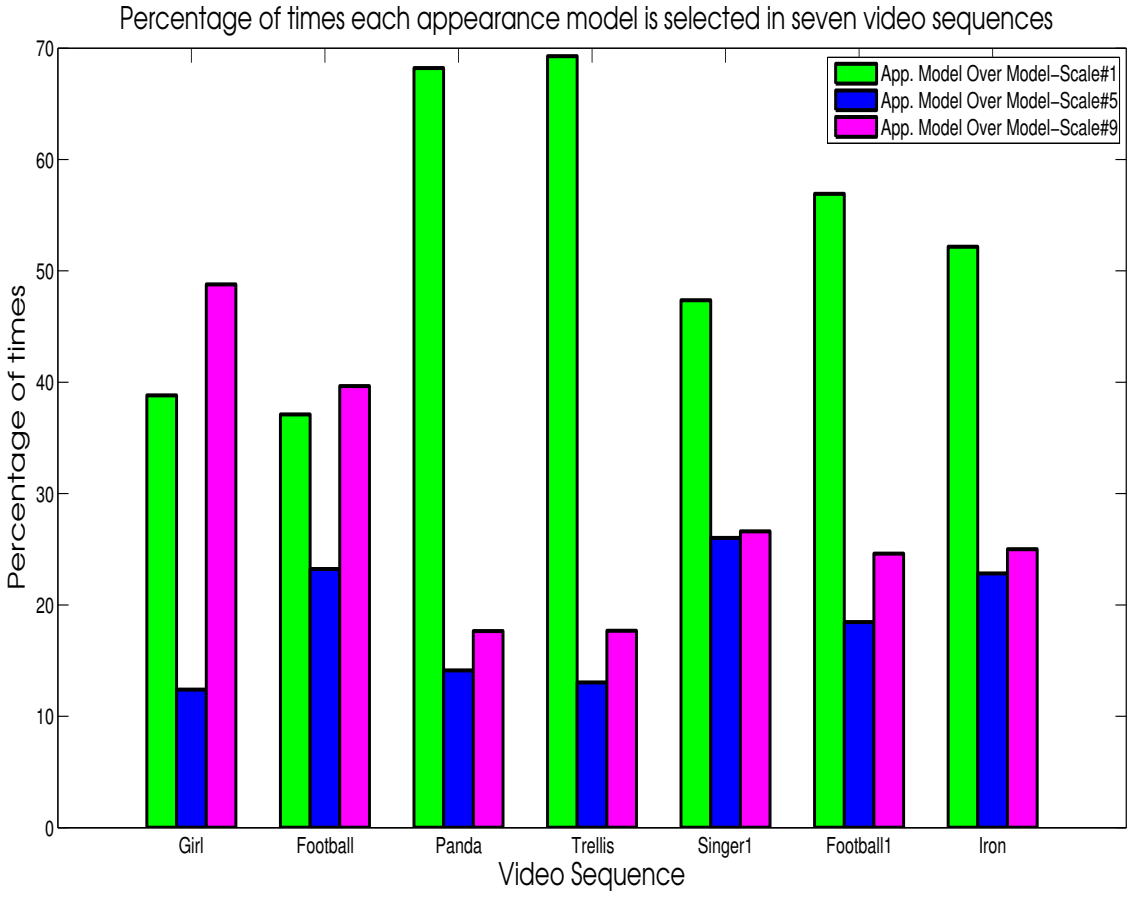


FIGURE 3.9: **Selection percentage for each appearance model in seven video sequences.** The figure illustrates percentage of times each appearance model is selected in seven video sequences.

Fig. 3.7(b) reveals appearance prediction performance in the *panda*, and *trellis* sequences. The ranks of individual models are almost the same in both the sequences. In contrast to the plots in Fig. 3.7(a), the plots in Fig. 3.7(b) reveal that the appearance model derived from model-scale 5 takes the lead in terms of performance over the appearance model corresponding to model-scale 9 after approximately one-third of both the sequences. This could be because the variations in most of the variables of the histogram are more or less unpredictable, i.e erratic in nature. To demonstrate, the temporal evolution of bin values in the first 200 frames of the *panda* sequence are displayed in the top row of Fig. 3.8(b), and the cumulative Bhattacharyya distance of appearance models corresponding to model-scale 5, and 9 is plotted in the bottom row of Fig. 3.8(b). It can be seen that the appearance model over model-scale 5 is a little superior to the appearance model derived from model-scale 9 under such variations. Similar behaviour can be observed in the *singer1* sequence in Fig. 3.7(c), in which the appearance model over model-scale 5 is a little better than the model over model-scale 9.

It was observed that appearance prediction performance improves when a set of appearance models, derived from multiple model-scales, is utilized in comparison to any individual model, but model over model-scale 1 contributes more to this improved performance than the others (learned models over model-scale 5, and 9) in almost all sequences. Since the learned models do not take into account the correlation among the bin values of the histogram, which is a strong assumption, they are not highly effective, and hence often their predicted appearances do not match closely to the true target appearance than the model over model-scale 1. A more powerful learning mechanism may remedy this.

Fig. 3.9 plots the percentage of times each appearance model is selected in seven video sequences. The ranks of appearance models over model-scales 1, and 9 in the *girl* and *football* sequences in Fig. 3.9 are the same when compared to the cumulative error plots of Fig. 3.7(a). However, the ranks of appearance models corresponding to model-scales 5, and 9 in the rest of the sequences are inverted in Fig. 3.9 when compared to the cumulative error plots of Fig. 3.7.

3.3 Conclusion

This chapter posed the question, is there any benefit in terms of prediction performance to be gained by using a set of models, derived from multiple temporal scales, rather than any fixed, individual model? This is the first step towards determining the potential of the idea of visual tracking over multiple temporal scales.

To analyze the benefits of learning over multiple temporal scales, both motion and appearance variations were investigated in the ground truth data of several challenging sequences. In general, the experimental results revealed that prediction performance improves with access to a set of models, provided it is possible to select which is the best model to use at any given time. For appearance prediction, this gain in the performance is not substantial when compared to the location prediction, but it is encouraging enough to develop this idea further in the future.

Some interesting results were also obtained from the individual performances of models. In situations where variations in motion or appearance form approximately regular patterns of some duration, the longer scale models perform better than the shorter scale models. On the other hand, when the variations are more erratic, the shorter scale models surpass their longer scale counterparts. Again, these performance gains are more notable in case of location prediction than appearance prediction.

Having seen the benefits of access to a set of multi-scale models, in the next chapter we will address the other two questions with respect to motion modelling: first, can we use these models, derived from multiple histories of the target state, to make flexible predictions multiple time-points ahead and overcome occlusions? And secondly: how can we automatically select the most suitable model at each time-point and fit this selection process into a tracking framework? We focus on motion modelling rather than appearance modelling as this fits more readily into a tracking framework than multi-scale appearance modelling, which requires further development.

Chapter 4

A Visual Tracker Operating Over Multiple Temporal Scales

In the previous chapter, it was observed that learning over multiple temporal scales has the potential to improve prediction performance. This is considered crucial in predictive tracking. A good motion prediction (close to the true target state) can improve the sampling efficiency of the search method and may make it more robust to local optima.

To address questions (2) and (3) that arose from the proposed idea of visual tracking over multiple temporal scales at the start of foregoing chapter, this chapter proposes a visual tracker operating over multiple temporal scales that is capable of handling occlusion and non-constant target motion. This is achieved by learning motion models from the target history at different temporal scales and applying those models over multiple temporal scales in the future. These motion models are learned online in a computationally inexpensive manner. To provide reliable recovery of tracking after occlusions, the bootstrap particle filter is extended to propagate particles at multiple temporal scales, possibly many frames ahead, guided by these motion models. In terms of Bayesian tracking, the prior distribution at the current time-step is approximated by a mixture of the most likely modes of several previous posteriors propagated using their respective motion models. This improved and rich prior distribution, formed by models learned and applied over multiple temporal scales, makes the proposed tracker more robust to complex target motion by covering a relatively large search space. Experiments have been carried out on both publicly available benchmarks and new video sequences. Results reveal that the proposed method successfully handles occlusions and a variety of rapid changes in target motion.

4.1 Introduction

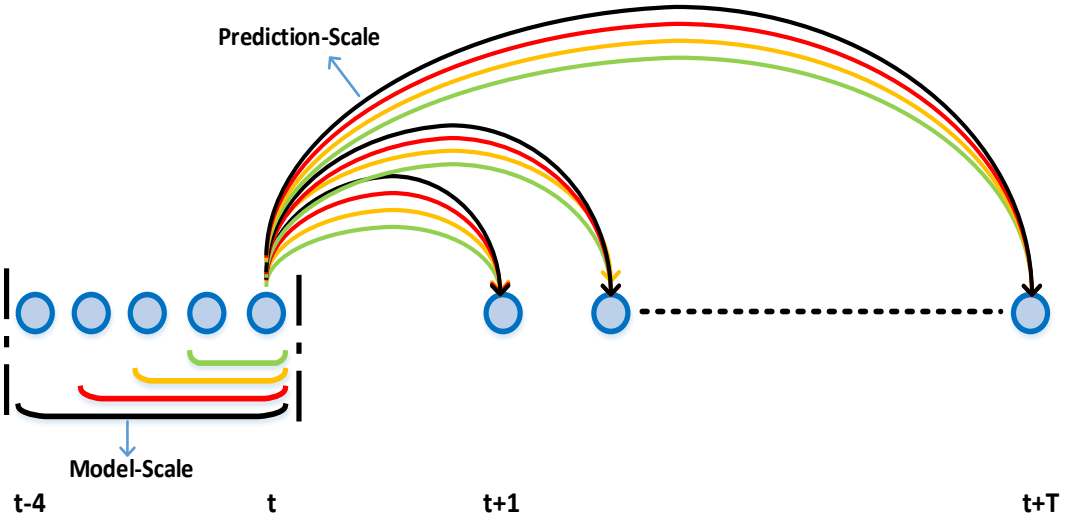
Occlusion is a complicated problem for visual tracking frameworks. In real-world scenarios, a tracked target can stay partially or fully occluded for variable time periods. Because the target image evidence is partially or wholly unavailable, it becomes hard for a tracker to maintain contact with the target through occlusions. This is particularly problematic if a change in the appearance or direction of motion occurs while the target is occluded. While many solutions have been proposed to the occlusion problem, it remains an open issue. For a thorough review of the existing work on occlusion handling, please refer to chapter 2 of this thesis.

Maintaining accurate estimates of a target's state is also difficult when it exhibits complex motion patterns. These can be the result of rapidly varying motion of a target, camera movement, and/or low frame rate of the video. As motion uncertainty increases during quickly varying movement, the search space of the parameters to be estimated during tracking becomes large. Although poor motion prediction can directly affect tracking accuracy under these situations, little attention has been paid to this problem. For a detailed study of the existing body of work on motion variations, please see chapter 2 of this thesis.

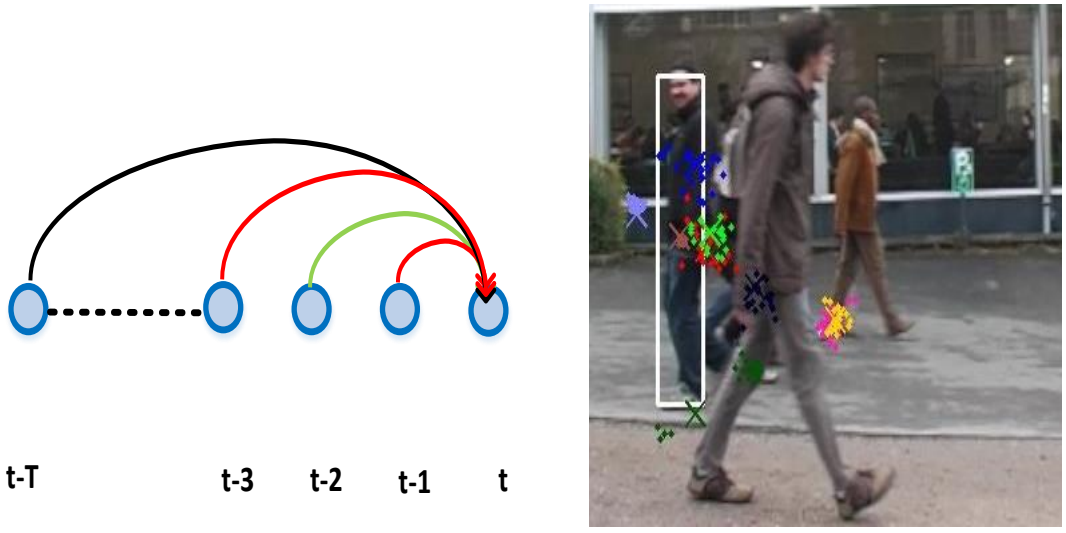
This chapter proposes a tracking framework exploiting models learnt and applied over multiple temporal scales that is capable of implicitly coping with occlusions and non-constant target motion without using strong appearance models, explicit occlusion detection mechanisms or exhaustive search methods.

To recover from occlusion a flexible prediction method is employed, which estimates target state at temporal scales up to the expected maximum duration of likely occlusions. To achieve this, motion models are learnt at multiple model-scales and used to predict possible target states at multiple prediction-scales ahead in time. The model-scale is the duration of a sequence of recently estimated target states over which a motion model is learnt. The prediction-scale is the temporal distance, measured in frames of the input image sequence, over which a given prediction is made. Reliable recovery of tracking after occlusions is achieved by extending the bootstrap particle filter to propagate particles to multiple prediction-scales, using models learnt at multiple model-scales. Fig. 4.1 summarises the approach.

The proposed framework can handle variable motion well due to the following: in predictive tracking, learnt motion models describe the recent history of target state —the most recent section of the target's path across the image plane. Trackers using, for example, a single linear motion model effectively represent the target path as a straight



(A) Multiple motion models are learned from the recent history of estimated states at different temporal scales, and each model is applied to multiple temporal scales in the future.



(B) This means that, when determining target state, multiple sets of motion models are available to make predictions. Each set includes models learnt at multiple model-scales. In the proposed method one model per set is selected to propagate particles.

FIGURE 4.1: **Visual Tracking Over Multiple Temporal Scales.**

line. By building multiple motion models at multiple model-scales, the proposed framework maintains a much richer description of target path. The diverse set of models produced captures at least some of the complexity of that path and, when used to make predictions, the model set represents variation in target motion better than any single model. Furthermore, when such models operate over multiple prediction-scales, they facilitate the development of a rich and improved prior distribution at each time-point.

This lets the proposed framework capture an increased search space, which is valuable in the presence of abrupt variations in target motion.

Tracking methods bearing resemblance to the proposed method are graph-based tracking methods working on temporal windows [Poiesi and Cavallaro, 2015, Shafique and Shah, 2005, Shu et al., 2012]. These methods formulate the multi-target tracking problem as a graph in which nodes corresponds to the detection responses and edges represent the cost of moving from one node to another. Given this graph, the aim is to produce several subgraphs in which the detections belonging to same object are connected [Zamir et al., 2012].

The detections are produced by a background subtraction algorithm or an object detector in each frame of a video sequence. To deal with occlusions and missed detections, these methods maintain a temporal buffer in which the correspondence problem is solved using optimization algorithms that compute approximate graph solutions [Poiesi and Cavallaro, 2015]. For instance, [Shafique and Shah, 2005] poses multi-frame correspondence problem in terms of a graph and proposes a non-iterative greedy algorithm for approximating the solution. [Poiesi and Cavallaro, 2015] recursively associated detections using a graph-based tracker on temporal windows and computed solution to the graph using a greedy algorithm. Short tracks are generated by optimally associating detections and the long tracks are formed by sequentially linking short tracks. The proposed approach also maintains temporal windows to overcome occlusions, however, it is different to aforementioned works in the following ways.

The proposed approach doesn't detect target in each frame and therefore doesn't link those detections in temporal windows to achieve tracking. It jointly estimates the target detection and track using state predictions (temporal priors) and particle clusters propagated from several previous time-points. These temporal priors are produced by motion models constituted over multiple model-scales at each time-point. A motion model selection mechanism selects the most suitable motion model (corresponding to a temporal prior) from each of the previous time-points. Above-mentioned approaches formulate the correspondence (linking) problem in terms of a graph and employ greedy optimization algorithms to compute the solution i.e. estimate trajectories of given targets. The proposed approach uses selected model from each previous time-point to propagate the most probable mode of the corresponding density to the current time-point to achieve tracking.

The remainder of this chapter is organized as follows. The proposed approach is expressed in terms of Bayesian tracking in Section 4.2. After this theoretical segment, implementation details of the proposed tracking framework are presented in Section 4.3. Then, the proposed framework is compared with competing methods on some publicly

available benchmarks and new video sequences, and analyzed in Section 4.5. Finally, Section 4.6 concludes the chapter, and outlines the potential limitations of the different components of the proposed framework, which will allow us to explore this method further in chapter 5.

4.2 Bayesian Tracking Formulation

The aim is to find the best state of the target at time t given observations up to time t. State at time t is given by $\mathbf{X}_t = \{X_t^x, X_t^y, X_t^s\}$,where X_t^x,X_t^y, and X_t^s represent the x,y location and scale of the target, respectively. In a Bayesian formulation, the proposed solution to tracking problem comprises two steps: update 4.1, and prediction 4.2.

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto p(\mathbf{Y}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{Y}_{1:t-1}). \tag{4.1}$$

where $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ is the posterior probability given the state $\mathbf{X}_t$ at time t, and observations $\mathbf{Y}_{1:t}$ up to t. $p(\mathbf{Y}_t|\mathbf{X}_t)$ denotes the observation model.

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t-1}) = \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})d\mathbf{X}_{t-1}. \tag{4.2}$$

where $p(\mathbf{X}_t|\mathbf{Y}_{1:t-1})$ is the prior distribution at time t, and $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is a motion model.

In this work, the accuracy of the posterior distribution at a given time t is improved by improving the prior distribution. Here, the prior distribution is approximated by a mixture of the most probable modes of T previous posteriors propagated by the T selected motion models, which are generated using information from up to T frames ago. Eq. 4.2 in the standard Bayesian formulation can now be written as:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t-1}) \approx \int_{k=1}^{k=T} p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k})dk, \tag{4.3}$$

which is now the sum of T individual predictive distributions originated from the T previously estimated posteriors.

$$p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}) \approx \int_{\mathbf{X}_{t-k}} p_k(\mathbf{X}_t|\mathbf{X}_{t-k})p(\tilde{\mathbf{X}}_{t-k})d\mathbf{X}_{t-k}. \tag{4.4}$$

where $p_k(\mathbf{X}_t|\mathbf{X}_{t-k})$ is the motion model selected at time t from a set of motion models learned at time $t-k$, and $p(\tilde{\mathbf{X}}_{t-k}) \subset p(\mathbf{X}_{t-k}|\mathbf{Y}_{1:t-k})$ is the most probable mode (approximated via N particles) of the posterior at time $t-k$. A relatively rich and improved prior distribution in Eq. 4.3 allows handling occlusions and abrupt motion variation in

a simple manner without resorting to complex appearance models and exhaustive search methods.

The posterior corresponding to the k_{th} predictive distribution can be written as:

$$p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}, \mathbf{Y}_t) \propto p(\mathbf{Y}_t|\mathbf{X}_t)p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}). \tag{4.5}$$

Then, the posterior distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ at time t is a sum of T individual posteriors:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) = \sum_{k=1}^{k=T} p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}, \mathbf{Y}_t). \tag{4.6}$$

When particles are used to approximate the k_{th} predictive distribution, which is formed by convoluting $p(\tilde{\mathbf{X}}_{t-k})$ with $p_k(\mathbf{X}_t|\mathbf{X}_{t-k})$, Eq. 4.4 becomes:

$$p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}) \approx \sum_{i=1}^{N} p_k(\mathbf{X}_t|\mathbf{X}_{t-k}^{(i)})p(\tilde{\mathbf{X}}_{t-k}^{(i)}). \tag{4.7}$$

Now the posterior corresponding to the approximated k_{th} predictive distribution can be written as:

$$p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}, \mathbf{Y}_t) \approx p(\mathbf{Y}_t|\mathbf{X}_t) \sum_{i=1}^{N} p_k(\mathbf{X}_t|\mathbf{X}_{t-k}^{(i)})p(\tilde{\mathbf{X}}_{t-k}^{(i)}). \tag{4.8}$$

The approximated posterior $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ at time t through particles is a sum of T individual posteriors:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto \sum_{k=1}^{k=T} p(\mathbf{Y}_t|\mathbf{X}_t) \sum_{i=1}^{N} p_k(\mathbf{X}_t|\mathbf{X}_{t-k}^{(i)})p(\tilde{\mathbf{X}}_{t-k}^{(i)}). \tag{4.9}$$

The best state of the target $\hat{\mathbf{X}}_t$ is obtained using Maximum a Posteriori (MAP) estimate over the $N_t = N \times T$ weighted particles which approximate $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$,

$$\hat{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t^{(m)}} p(\mathbf{X}_t^{(m)}|\mathbf{Y}_{1:t}) \; for \; m = 1, ..., N_t, \tag{4.10}$$

where $\mathbf{X}_t^{(m)}$ is the m_{th} particle.

4.3 A Multiple Temporal Scale Framework

The core idea is to construct an improved and rich prior distribution at each time-point by combining sufficient particle sets that at least one set will be valid and allow recovery from occlusion and robustness to non-constant motion. A valid particle set is the most likely mode of an accurate estimation of the posterior probability from some previous time-point, propagated by a motion model generated over an appropriate model-scale and unaffected by occlusion. This is in contrast to most existing approaches [Isard and Blake, 1998a],[Ross et al., 2008],[Pérez et al., 2002],[Bao et al., 2012], and [Jia et al., 2012], in which the posterior from the most recent time-point $(t-1)$ is propagated to the current time-point (t).

In the proposed method, multiple *sets* of motion models are available at each time-point. The models in each set are all learned at a single previous time-point, but over multiple model-scales. One motion model is selected from each set, and used to propagate particles forward from the time at which it was learned. Propagation from several adjacent time-points using selected motion models generates several particle sets at each future time-point, within a certain temporal distance.

At each time t, the proposed algorithm proceeds through three stages: evaluation, learning, and prediction.

4.3.1 Evaluation

T sets of motion models are available at time t, one from each of the T preceding time-steps. Each set of models at time t is represented by its corresponding set of predictions. Based on these predictions, the most suitable motion model from each set is selected, and used to generate particles describing target state at time t.

The T sets of motion models available at time t are represented by the corresponding T sets of predicted states at time t. Let $L_t = \{\mathbf{l}_t^k | k = 1, ..., T\}$ denote T different sets of states predicted by their respective motion models, where $\mathbf{l}_t^k = \{l_t^{j,k} | j = 1, ..., G\}$ is a set of states predicted by G motion models belonging to (learned at) the k_{th} previous time-step. $l_t^{j,k}$ denotes the predicted state by j_{th} motion model learned at k_{th} previous time-step. For instance in Fig. 4.2(a), $\mathbf{l}_t^1$ is a set containing 4 states predicted by 4 motion models learned at time $t-1$.

4.3.1.1 Model Set Reduction

The aim of model set reduction is to establish search regions for the particle filter in which there is a high probability of the target being present. This in turn will reduce the sampling effort, as search regions corresponding to all the predictions no longer need to be searched.

The most suitable motion model $\mathbf{R}_t^k$ is selected from each set using the following criterion on the corresponding set of predicted states $\mathbf{l}_t^k$:

$$\hat{l}_t^k = \arg\max_{l_t^{j,k}} p(\mathbf{Y}_t|l_t^{j,k}) \tag{4.11}$$

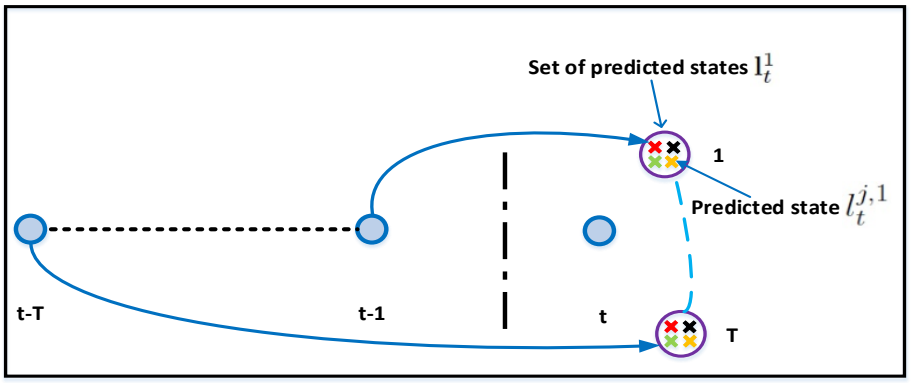
where $\hat{l}_t^k$ is the most suitable state prediction from the set $\mathbf{l}_t^k$, and $p(\mathbf{Y}_t|l_t^{j,k})$ measures the visual likelihood at the predicted state $l_t^{j,k}$. In other words, $\hat{l}_t^k$ is the most suitable state prediction of the most suitable motion model $\mathbf{R}_t^k$. For example, Fig. 4.2(b) shows the predicted state $\hat{l}_t^1$ of the most suitable motion model $\mathbf{R}_t^1$ chosen from 4 motion models learned at time $t-1$. After this selection process, the T sets of motion models are reduced to T individual models.

One motion model is selected at time t from each of the T previous time-points because it is assumed that at least one of them would have generated an accurate model. An accurate model is one whose state prediction at time t is close to the true target state. It is of course possible that none of the models belonging to one or more previous time-points are accurate enough.

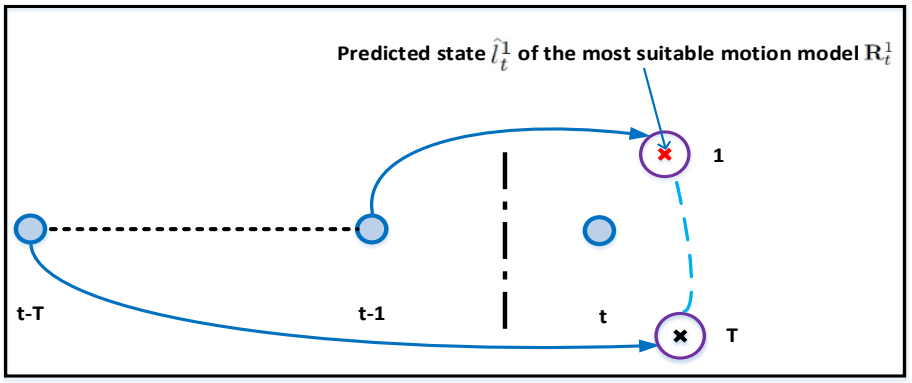
4.3.1.2 Propagation of Particles

In the bootstrap particle filter [Arulampalam et al., 2002], the posterior probability at time $t-1$ is estimated by a set of particles $\mathbf{X}_{t-1}^{(i)}$ and their weights $\omega_{t-1}^{(i)}$,$\{\mathbf{X}_{t-1}^{(i)}, \omega_{t-1}^{(i)}\}_{i=1}^N$, such that all the weights in the particle set sum to one. The particles are resampled to form an unweighted representation of the posterior $\{\mathbf{X}_{t-1}^{(i)}, 1/N\}_{i=1}^N$. At time t, they are propagated using the motion model $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ to approximate a prior distribution $p(\mathbf{X}_t|\mathbf{Y}_{t-1})$. Finally, they are weighted according to the observation model $p(\mathbf{Y}_t|\mathbf{X}_t)$, approximating the posterior probability at time t.

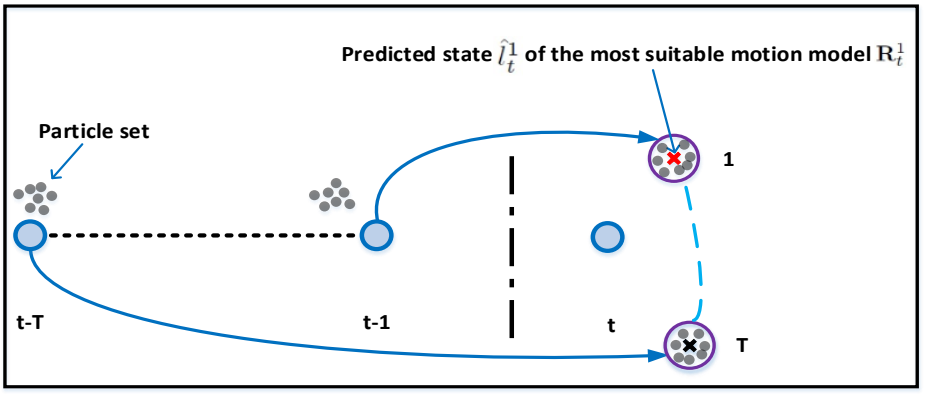
Here, particle sets not just from one previous time-step $(t-1)$, but from T previous time-steps are propagated to time t using the T selected motion models. When using first-order polynomial (linear) motion models the most suitable motion model $\mathbf{R}_t^k$ selected from those learnt at the k_{th} previous time-step will propagate a particle set from the k_{th}



(A) In the evaluation stage, there exist T different sets of predicted states at time t, where each set $\mathbf{l}_t^k$ comprises G states predicted by G motion models learned at k_{th} previous time-step. In this figure, $\mathbf{l}_t^1$ is a set composed of 4 states predicted by 4 motion models learned at time $t-1$.



(B) **Model Set Reduction.** T sets of motion models available at time t, represented by the corresponding T sets of predicted states, are reduced to T individual models. This is achieved by selecting the most suitable motion model $\mathbf{R}_t^k$ from G motion models learned at k_{th} previous time-step. This figure shows the predicted state $\hat{l}_t^1$ of the most suitable motion model $\mathbf{R}_t^1$ selected from 4 motion models learned at time $t-1$.



(C) **Propagation of Particles.** T selected motion models, one from each of the T preceding time-steps, are used to propagate particle sets from T preceding time-steps to time t. In this figure, the most suitable motion model $\mathbf{R}_t^1$, selected from 4 motion models learned at time $t-1$, and represented by its predicted state $\hat{l}_t^1$, is used to propagate particle set from time $t-1$ to time t.

FIGURE 4.2: **Graphical illustration of events occurring at the evaluation stage.**

previous time-step as follows

$$X^x_{t,k} = X^x_{t-k} + g(\mathbf{R}^k_t)k + \mathcal{N}(0, \sigma^2_x k), \tag{4.12}$$

where X^x is the horizontal part of the target state, $g()$ indicates the slope of the model, and $\mathcal{N}(0, \sigma_x)$ is a Gaussian distribution with zero-mean and σ^2_x variance. For instance, in Fig. 4.2(c), the most suitable motion model $\mathbf{R}^1_t$, is used to propagate a particle set from time $t-1$ to time t.

Propagation from the last T time-steps, generates T particle sets at time t. Since the prior distribution at time t is now a combination of T predictive distributions (particle sets), it is rich compared to the prior of the original bootstrap particle filter. Each propagated particle set (k_{th} predictive distribution $p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k})$) is now weighted according to the observation model $p(\mathbf{Y}_t|\mathbf{X}_t)$ to form the respective posterior $p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}, \mathbf{Y}_t)$. Then, the final posterior $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ at time t is a sum of T individual posteriors.

If the target was occluded for less than or equal to $T-1$ frames, it may be recovered by a set of particles unaffected by the occlusion. To focus on particles with large weights, and reduce computational cost, the first N particles are retained after the resampling step.

The best state of the target $\hat{\mathbf{X}}_t$ is obtained using MAP estimate over the $N_t = N \times T$ weighted particles which approximate $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ (Eq.4.10).

4.3.1.3 Model (Prediction) Status

It is important to mention the number of motion models available in the evaluation stage at time-points ahead of the current time t since this is not equal everywhere. T time-points ahead of time t will be considered because the proposed framework operates at T temporal scales.

There are $G \times (T-1)$ predicted states (or $T-1$ sets of predicted states) at time $t+1$, and $G \times 0$ predicted state (or 0 set of predicted state) at time $t+T$. In other words, there are $G \times (T-1)$ motion models (or $T-1$ sets of motion models) at time $t+1$, and $G \times 0$ motion model (or 0 set of motion models) available at time $t+T$ (See Fig. 4.3). Recall that the G is the cardinality of the set of motion models learnt at each time-point.

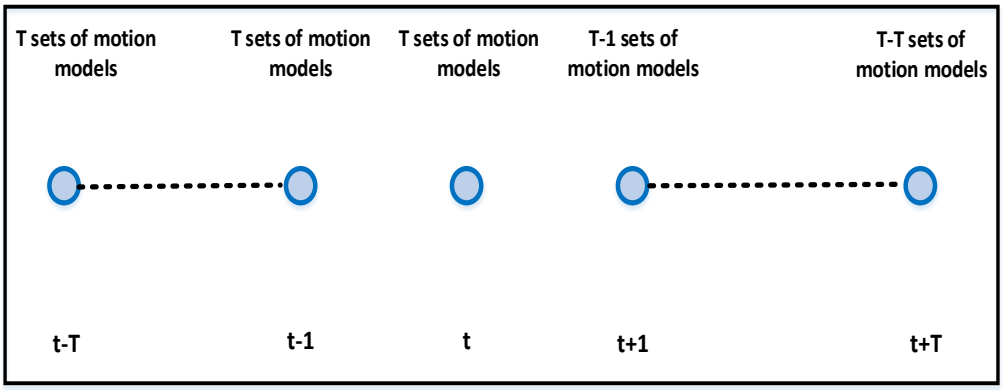


FIGURE 4.3: **Model Status at time-points ahead of time t (current time-point) in the evaluation stage.** $T-1$ sets of motion models are present at time $t+1$, and $T-T$ set of motion models is available at time $t+T$ when proceeding through the evaluation stage.

4.3.2 Learning

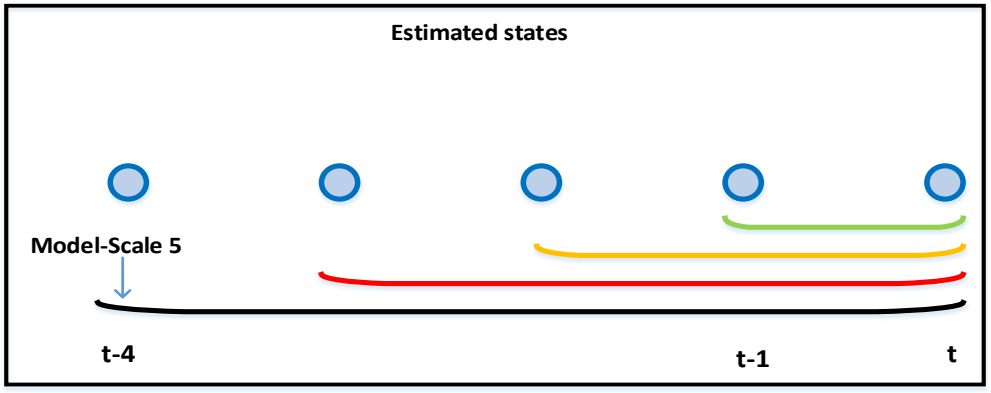
Having estimated the target state at time t, the task of this stage is to learn linear motion models at multiple model-scales that can be used to make forward predictions. A linear motion model is represented by $\mathbf{M}$. $\mathbf{M}$ is learned at a given model-scale separately for the x-location, y-location, and scale s of the target's state. Fig. 4.4(a) shows four linear motion models learned over four different model-scales at time t.

Here, a linear motion model over model-scale m is learned, taking into account how well the states in this sequence have been estimated and their relevance in terms of how far each estimated state is from the most recently estimated state [Kristan et al., 2010]. Let $Z^m = \{\hat{x_n}\}_{n=t-m+1}^{n=t}$ represent a sequence of recently estimated x-components of target states over model-scale m, and let $Q^m = \{\hat{\theta_n}\}_{n=t-m+1}^{n=t}$ be a set of their weights. The linear motion model over model-scale m at time t can be written as:

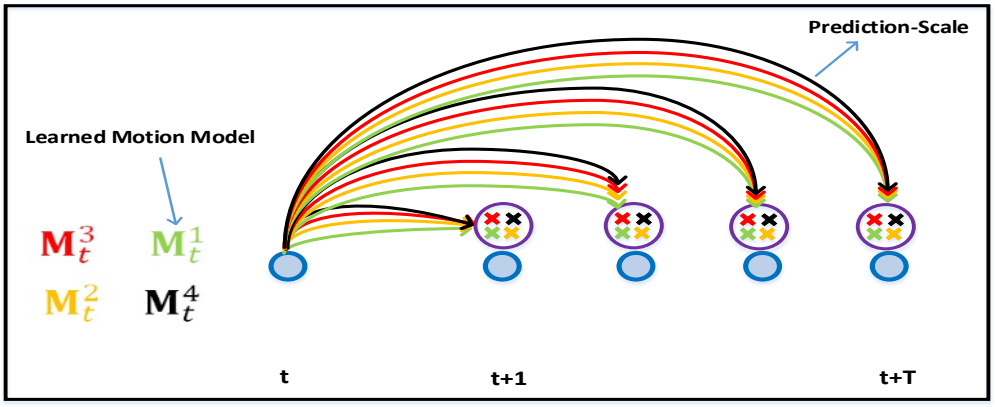
$$\tilde{x_n} = \phi_{\hat{x_t}}^m + \tau_{\hat{x_t}}^m n, \tag{4.13}$$

where τ is the slope, ϕ the intercept, and $\hat{x_t}$ denotes that the model parameters have been learnt using a sequence of recently estimated x-components of target states whose last member is $\hat{x_t}$. Eq. 4.13 is same as Eq. 3.1 and it is repeated here in an effort to make the chapter self-contained. Now the parameters, $\phi_{\hat{x_t}}^m$ and $\tau_{\hat{x_t}}^m$, in Eq. 4.13 can be determined via weighted least squares (WLS) method by minimizing the following weighted sum of squared differences [Kristan et al., 2010]

$$S(\phi_{\hat{x_t}}^m, \tau_{\hat{x_t}}^m) = \sum_{n=t-m+1}^{t} \mathcal{M}_t^{(n)} (\hat{x_n} - \tilde{x_n})^2, \tag{4.14}$$



(A) In the learning stage, multiple motion models are constituted at multiple model-scales using the recent history of estimated states at time t. In this figure, four linear motion models are learned over four different model-scales at time t. The four model-scales are 2,3,4, and 5.



(B) At the prediction stage, a set of learned motion models are used to predict possible target states at T prediction-scales. In this figure, a set comprising four learned motion models is shown at time t. Each motion model predicts possible target state at T prediction-scales.

FIGURE 4.4: **Graphical illustration of the learning stage and the prediction stage.**

where $\mathcal{M}_{(.)}^{(n)}$ are the weights defined as

$$\mathcal{M}_t^{(n)} = \hat{\theta}_n \mathrm{e}^{-0.5\frac{(n-t)^2}{\sigma_o^2}}. \tag{4.15}$$

The first term in Eq. 4.15 is the visual likelihood score at $\hat{x_n}$, and the second term is a Gaussian. A Gaussian function is used to attenuate the importance of farther states as recent states are more pertinent in explaining targets current motion [Kristan et al., 2010]. Note that other functions that show similar behaviour (e.g., an exponential function) can also be used in place of a Gaussian [Kristan et al., 2010]. Effectively the $m = 3\sigma_o$ most recently estimated states are considered in Eq. 4.14 because the weights

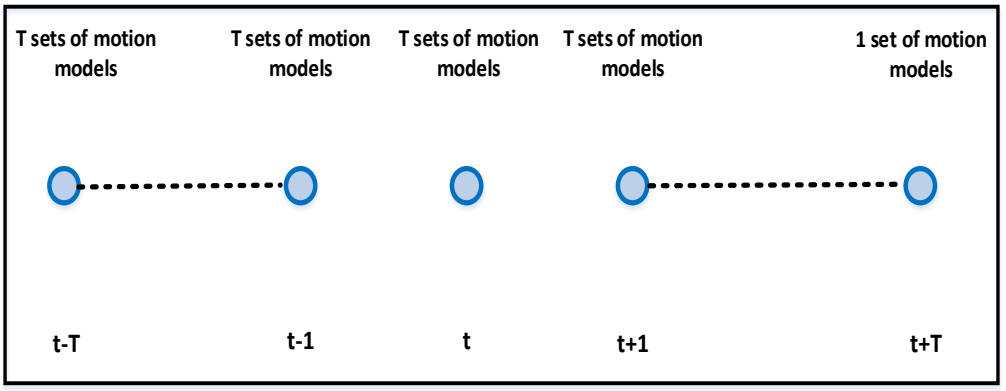


FIGURE 4.5: **Model Status at time-points ahead of time t (current time-point) after the prediction stage.** T sets of motion models are present at time $t+1$, and 1 set of motion models is available at time $t+T$ after the prediction stage.

of all other states are very small.

4.3.3 Prediction

Learned motion models are now used to predict possible target states at T prediction-scales. Let $\mathbf{M}_t^{j=1,\ldots,|\mathbf{M}_t|}$ represent a set of learned motion models at time t, where $|.|$ is the cardinality of the set. The cardinality of this set is G, and each model predicts target state $l(\tilde{x}, \tilde{y}, \tilde{s})$ at T prediction-scales. For example, Fig. 4.4(b) shows a set of $G = 4$ learned motion models at time t predicting possible target states. The proposed method is summarized in Algorithm 1.

4.3.4 Model (Prediction) Status

It is worth mentioning here the available number of models (predictions) at time-points ahead of time t after the prediction stage.

There would be $G \times T$ predicted states (or T sets of predicted states) at time $t+1$, and G predicted states (or 1 set of predicted states) at time $t+T$. In other words, there would be $G \times T$ motion models (or T sets of motion models) at time $t+1$, and G motion models (or 1 set of motion models) available at time $t+T$ (See Fig. 4.5).

In comparison to the model status in the evaluation stage, here, the set(s) of models (predictions) available at each time-point ahead of time t (is) are increased by 1. This is because now the models have been learnt at time t, and they have generated state predictions over T prediction-scales.

Algorithm 1 A Multiple Temporal Scale Tracker

Input: $\mathbf{P}_t = \emptyset$

Let $\mathbf{W} = \{W_{t-1}, ..., W_{t-T}\}$ represent the resampled sets of particles after estimation of the posterior from T previous time-steps, where $W_{t-1} = \{\mathbf{X}_{t-1}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$.

Output: Best state $\hat{\mathbf{X}}_t$ at time t.

Evaluation Stage

for $k = 1$ *to* T

for $j = 1$ *to* G

- Measure visual likelihood $p(\mathbf{Y}_t|l_t^{j,k})$, where $l_t^{j,k}$ is the predicted state at time t by j_{th} motion model from k_{th} previous time-step.

end

- Select the most suitable motion model $\mathbf{R}_t^k$ at time t using Eq. 4.11.
- Approximate k_{th} predictive distribution $p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k})$ by propagating the particle set from k_{th} previous time-step $W_{t-k} = \{\mathbf{X}_{t-k}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$ using Eq. 4.12 by taking the slope of selected motion model $\mathbf{R}_t^k$ to time t. After propagation the particle set is represented by $W_t^k = \{\mathbf{X}_t^{(i)}, \frac{1}{N}\}_{i=1}^{N}$.
- Convert k_{th} predictive distribution into the respective posterior distribution $p_k(\mathbf{X}_t|\mathbf{Y}_{1:t-k}, \mathbf{Y}_t)$ using $p(\mathbf{Y}_t|\mathbf{X}_t)$. After weighting the particle set is represented by $W_t^k = \{\mathbf{X}_t^{(i)}, \omega_t^{(i)}\}_{i=1}^{N}$.
- $\mathbf{P}_t = \mathbf{P}_t +\!\!+ W_t^k$. ($+\!\!+$ is the concatenation symbol)

end

- $\mathbf{P}_t$ represents the approximated posterior $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ at time t.
- Estimate the best state $\hat{\mathbf{X}}_t$ using Eq 4.10.
- Retain the first N particles $W_t = \{\mathbf{X}_t^{(i)}, \frac{1}{N}\}_{i=1}^{N}$ after resampling $\mathbf{P}_t$.

Learning Stage

for $e = 1$ *to* G

- $m = mdscales(e)$, where $mdscales$ is an array containing G model-scales.
- Learn linear motion model at m_{th} model-scale separately on locations x,y, and scale s of the target state.

end

Prediction Stage

for $e = 1$ *to* G

- Predict locations x,y, and scale s of the target state at T prediction-scales using Eq. 4.13.

end

4.4 Applying the proposed framework to the two-stage motion model

Algorithm 1 summarizes the proposed framework as applied to linear motion models. The proposed framework has also been applied to the two-stage model of [Kristan et al., 2010] to demonstrate its generality. The RW and NCV models present two extremes in the temporal correlation of velocity. The two-stage model is based on a more general approach, known as a Gauss-Markov process (GMP), that allows velocity to be modelled as correlated noise, but without taking into consideration the extent to which it is correlated. By modelling motion with a GMP, it is possible to capture dynamics which lie between RW and NCV. The state of the target now includes an additional internal velocity term v in both the x and y directions. With the two-stage model Eq. 4.12 becomes:

$$X^x_{t,k} = \begin{cases} X^x_{t-k} + \gamma_1 g(\mathbf{R}^k_t)k + \phi_{1,2} v^x_{t-k} + \Omega_x & k = 1 \\ X^{\acute{x}}_{t,k} + \Omega_x & k > 1 \end{cases} \tag{4.16a}$$

$$X^{\acute{x}}_{t,k} = X^{\acute{x}}_{t,k-1} + \gamma_1 g(\mathbf{R}^k_t) + \phi_{1,2} v^{\acute{x}}_{t,k-1} \tag{4.16b}$$

$$X^{\acute{x}}_{t,1} = X^x_{t-k} + \gamma_1 g(\mathbf{R}^k_t) + \phi_{1,2} v^x_{t-k} \tag{4.16c}$$

$$v^x_{t,k} = \begin{cases} \phi_{2,2} v^x_{t-k} + \gamma_2 g(\mathbf{R}^k_t) + \Omega_v & k = 1 \\ v^{\acute{x}}_{t,k} + \Omega_v & k > 1 \end{cases} \tag{4.16d}$$

$$v^{\acute{x}}_{t,k} = \phi_{2,2} v^{\acute{x}}_{t,k-1} + \gamma_2 g(\mathbf{R}^k_t) \tag{4.16e}$$

$$v^{\acute{x}}_{t,1} = \phi_{2,2} v^x_{t-k} + \gamma_2 g(\mathbf{R}^k_t) \tag{4.16f}$$

$$[\Omega_x, \Omega_v]^\intercal \sim \mathcal{N}(0, Q_x k) \tag{4.16g}$$

Where Ω_x and Ω_v are the noise processes acting on the target's position and the associated internal velocity (in this case it is x-part of the target's position), respectively. $\mathcal{N}(0, Q_x k)$ is the zero-mean Gaussian distribution with Q_x as the covariance matrix. $\phi_{1,2}$ and γ_1 are the proportions in which the the internal velocity v^x_{t-k} and the slope of the selected motion model $g(\mathbf{R}^k_t)$, also called rigid velocity, are combined into the deterministic part of the velocity acting on the position X^x_{t-k} at time $t-k$. Note that the nondeterministic part of the velocity acting on the position X^x_{t-k} is $\mathcal{N}(0, \sigma^2_x)$. Likewise, $\phi_{2,2}$ and γ_2 are the proportions in which the internal velocity v^x_{t-k} and the rigid velocity $g(\mathbf{R}^k_t)$ are combined into the deterministic part of the velocity acting on the velocity v^x_{t-k} at time $t-k$. For the full derivation of the aforementioned proportions[1], please refer to [Kristan et al., 2010].

[1]We set the correlation time parameter β of the two-stage to 10 in all the experiments. With this setting, the values of $\phi_{1,2}$,γ_1,$\phi_{2,2}$, and γ_2 become 0.1,0.9,10^{-5}, and 1, respectively.

Upon applying the proposed framework to the two-stage model, we have two ways to estimate target state: the rigid prediction $\check{l}_t = (\check{x}_t, \check{y}_t, \check{s}_t)$ and the flexible estimate $\hat{\mathbf{X}}_t = (\hat{X}_t^x, \hat{X}_t^y, \hat{X}_t^s)$. From the selected polynomial motion models $\mathbf{R}_t^{k=1:T}$ available at time t, the prediction of the model with the highest visual likelihood score is taken as the rigid prediction $\check{l}_t = (\check{x}_t, \check{y}_t, \check{s}_t)$. Eq. 4.16a propagates x and y components of the particles from the k_{th} previous time-step to time t by taking the slope of the selected polynomial motion model (in case of linear) as the rigid velocity, while the s component is propagated using Eq. 4.12. After propagation from T previous time-steps, the flexible estimate of the target state $\hat{\mathbf{X}}_t = \{\hat{X}_t^x, \hat{X}_t^y, \hat{X}_t^s\}$ is computed using Eq. 4.10. Now the normalized state of the target $\hat{n}_t$ is calculated by reducing the variance of the flexible estimate of the target state $\hat{\mathbf{X}}_t$ by fusing it with the rigid prediction $\check{l}_t$ of the target state:

$$\hat{n}_t = \frac{\check{l}_t \psi_{\check{l}_t} + \hat{\mathbf{X}}_t \psi_{\hat{\mathbf{X}}_t}}{\psi_{\check{l}_t} + \psi_{\hat{\mathbf{X}}_t}}, \tag{4.17}$$

where $\psi_{\check{l}_t}$ is the visual likelihood score at $\check{l}_t$, and $\psi_{\hat{\mathbf{X}}_t}$ is the visual likelihood score at $\hat{\mathbf{X}}_t$.

4.5 Experimental Details and Results

4.5.1 Data

To evaluate the accuracy of the proposed method, 3 different sets of sequences were compiled. In total, fourteen sequences were used. Ten are publicly available (*PETS 2001 Dataset 1*[2], *TUD-Campus*[Andriluka et al., 2008], *TUD-Crossing*[Andriluka et al., 2008], *Person*[Dihl et al., 2011], *car*[Wu et al., 2013], *jogging*[Wu et al., 2013], *deer*[Wu et al., 2013], *lemming*[Wu et al., 2013], *boy*[Wu et al., 2013] and *PETS 2009 Dataset S2*[3]) and four are our own (*squash*, *ball1*,*ball2*, and *toy1*). Fig. 4.6 shows the first frame of each sequence, with the target to be tracked annotated with a bounding box.

The selection of sequences for each set is carried out based on the chief criterion that a given sequence must involve occlusions (partial and/or full) of variable lengths and/or abrupt motion variation. This is because the proposed method aims at handling these two problems. Furthermore, the selection ensured that each set includes targets of different types (e.g. a ball, a person, or a car) to be tracked and a mix of indoor and outdoor environments.

The first set contains 9 sequences (*ball2*, *TUD-Campus*, *TUD-Crossing*, *Person*, *PETS 2001*, *PETS 2009*, *car*, *jogging*, and *toy1*) and each sequence contains occlusions (partial

[2] *PETS 2001 Dataset 1* is available from http://ftp.pets.rdg.ac.uk/
[3] *PETS 2009 Dataset S2* is available from http://www.cvg.rdg.ac.uk/PETS2009/

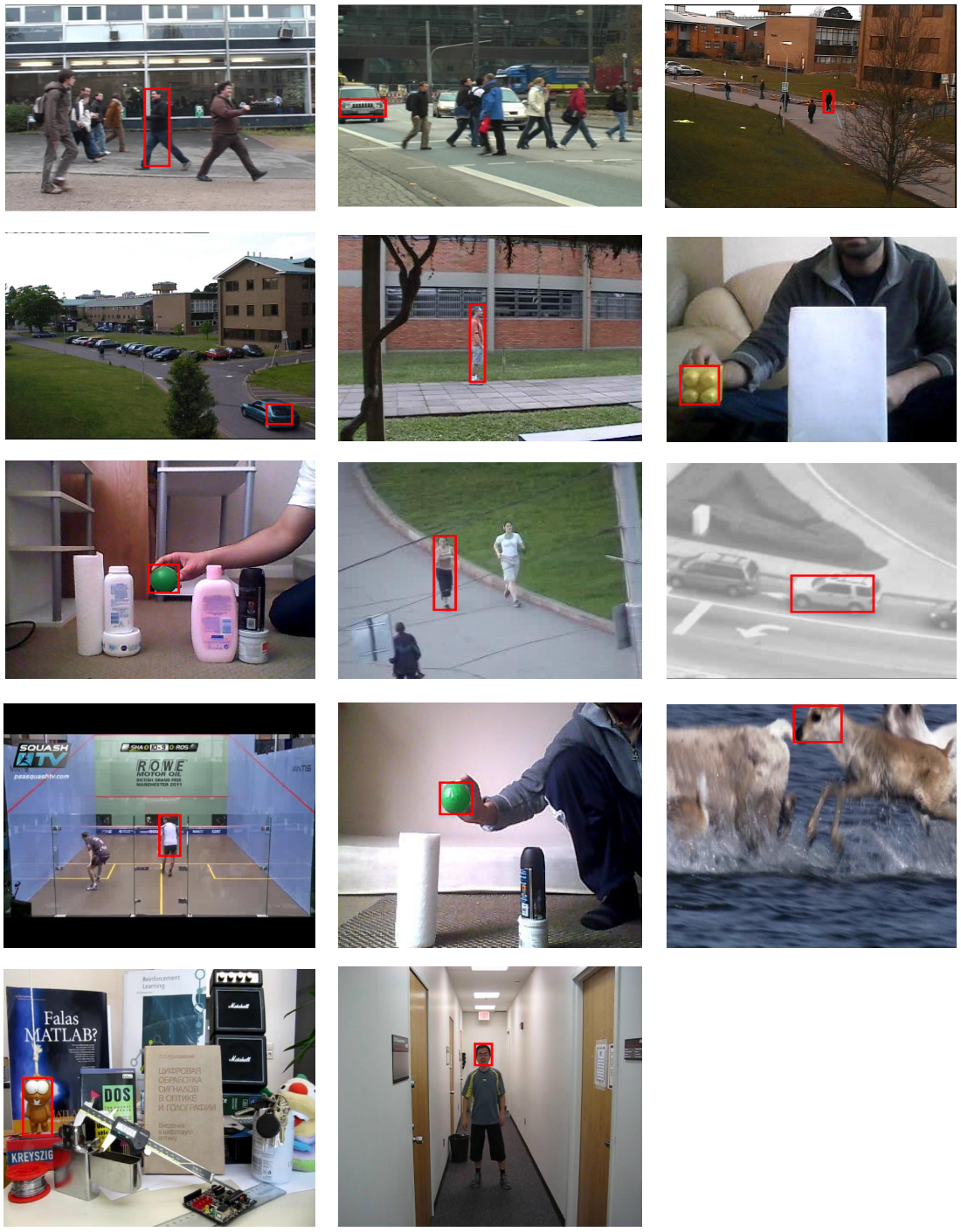


FIGURE 4.6: **Image sequences used for evaluation.** The first frame with the bounding box (red) of the target to be tracked is shown for each sequence.

and/or full) of different time periods. In addition, the targets in these sequences undergo scale variations and modest appearance changes due to illumination variations. The second set contains 2 sequences (*squash* and *ball1*) and the target in each sequence displays abrupt motion variations and faces occlusions of variable time periods. The last and the third set comprise 3 sequences (*deer*, *lemming*, and *boy*) and the target motion in each sequence undergoes abrupt variations. Along with non-constant target motion, these sequences include challenges such as background clutter and out-of-plane rotations.

4.5.2 Evaluation Protocol

We used three metrics for evaluation: center location error, Pascal score [Santner et al., 2010], and precision at a fixed threshold of 20 pixels [Babenko et al., 2011].

Among the many evaluation metrics used for tracking, center location error is most common. It is defined as the Euclidean distance between the center location of the tracked target and the manually labeled ground truth [Wu et al., 2013]. Then the center location error over all the frames of a video sequence is averaged to summarise performance for this sequence. This measure provides a broad indication of performance, but cannot identify how well the tracker maintained contact with the target throughout a video sequence. A tracker that closely captures the target for most of the sequence, but then fails completely on the last several frames, might for example have an average center location error higher than a tracker that follows the target throughout the course of the sequence, though not as precisely [Babenko et al., 2011].

For the above reason, [Babenko et al., 2011] adopted the precision plot. This reports the percentage of frames in which the estimated target location stays within a predefined threshold distance (in pixels) of the ground truth. In this work, precision is shown at a threshold distance of 20 pixels from the ground truth.

When a target exhibits scale and rotational changes, the precision metric cannot measure performance correctly. It only compares position part of the estimated target state and the corresponding ground truth. To overcome these problems, another evaluation metric known as the Pascal score [Santner et al., 2010] is used. Given the tracked bounding box b_t, and the ground truth bounding box b_g, the Pascal score is defined as $S = \frac{area(b_t \cap b_g)}{area(b_t \cup b_g)}$, where $\cap$, and $\cup$ represent the intersection and union of two regions, respectively. A frame is counted as successfully (correctly) tracked whose Pascal score S is above 0.5. So, the percentage of successfully tracked frames of a sequence summarizes performance on this sequence.

4.5.3 Experimental Settings

The appearance model used in all the experiments reported here was the colour histogram of [Pérez et al., 2002]. The aim is to investigate the power of multiple temporal scales to deal with occlusion and abrupt motion variation. To evaluate this independently of the appearance model, a simple appearance model is used on purpose. The Bhattacharyya coefficient was used as the distance measure. Linear motion models with model-scales of 2,3,4, and 5 frames were used (four models in total).

These model-scales were chosen empirically, but there are two reasons why these particular model-scales. The first and the primary reason for this is that we wanted to know when utilizing multiple temporal scales, if there are better models available than the zero-order model [Dellaert et al., 1999, Jia et al., 2012, Mei and Ling, 2009, Ross et al., 2008]. With this question, the appropriate direction was to explore the immediate longer scale models 2, 3, 4, and 5 for the range of model-scales. Lastly, the number of models i.e. 4 was chosen to have (include) different flavours of model-scales in the model set. It contains model-scales of shorter duration as well as of longer duration with reference to zero-order model. Of course more model-scales (of longer duration) can be added, but that might not add diversity to the model set.

MTS-L denotes the proposed method applied over a linear motion model (Algorithm 1). As mentioned earlier, we also apply our proposed framework to the two-stage model of [Kristan et al., 2010], which is denoted by MTS-TS, to show its generality. In MTS-TS, the β parameter of the two-stage model was fixed at 10, giving high weight to the rigid velocity, estimated by the linear motion model, and very low weight to the internal velocity. As a result, it becomes strongly biased towards the predicted location, but still allows some deviation.

The proposed method was compared to three baseline and seven state-of-the-art trackers. The first two baseline trackers, T_{RW} and T_{NCV}, were colour based particle filters from [Pérez et al., 2002], but use different motion models. T_{RW} used a random-walk model while T_{NCV} used a nearly constant velocity model. The third baseline tracker T_{TS} was the two-stage dynamic model proposed by [Kristan et al., 2010]. The parameters, K and β, in [Kristan et al., 2010] were set to 5 and 10, respectively.

The state-of-the-art trackers are SCM [Zhong et al., 2012], ASLA [Jia et al., 2012], L1-APG [Bao et al., 2012],VTD [Kwon and Lee, 2010], FragT [Adam et al., 2006], SemiBoost [Grabner et al., 2008], and WLMCMC [Kwon and Lee, 2008]. The minimum and maximum number of samples used for WLMCMC, VTD, SCM, ASLA, and L1-APG was 600 and 700, respectively. Our proposed tracker is implemented in MATLAB and

runs at about 3 frames/sec with 640 particles. Appendix A contains a list of tracker parameters used.

The state-of-the-art trackers were chosen keeping in view two important properties: their performance according to the CVPR'13 benchmark [Wu et al., 2013], and their ability to handle occlusions (partial and full) and abrupt motion variations. SCM and ASLA both have top ranked performance on the CVPR'13 benchmark. SCM combines a sparsity based classifier with a sparsity based generative model and incorporates an occlusion handling mechanism, while ASLA is based on a local sparse appearance model and is robust to partial occlusions. In L1-APG, the coupling of L1 norm minimization and an explicit occlusion detection mechanism makes it robust to partial as well as full occlusions. The integration of two motion models having different variances with a mixture of template-based object models lets VTD explore a relatively large search space, while remaining robust to a wide range of appearance variations. FragT was chosen because its rich, patch-based representation makes it robust to partial occlusion. SemiBoost was picked as it searches the whole image space once its tracker loses target, and thus, it can re-locate the target after full occlusions. WLMCMC searches the whole image space by combining an efficient sampling strategy with an annealing procedure that allows it to capture abrupt motion variations quite accurately and re-locate the target after full occlusions.

4.5.4 Comparison with competing methods

4.5.4.1 Quantitative Evaluation

Tables 4.1(a),(b) summarise tracking results obtained from image sequences in which the target is occluded. The numbers in table 4.1(a) indicate the center location error (in pixels) averaged over all frames of the sequence. In 5 out of 9 sequences, MTS-L tracked the target more accurately than the competing methods. MTS-L reliably recovered the target after severe occlusions by efficiently allocating particles at multiple prediction-scales through motion models learned over multiple model-scales. VTD performed badly in all 9 sequences because inappropriate appearance model updates during longer occlusions cause drift from which it cannot recover. Although SemiBoost uses explicit re-detection once the target is lost, its accuracy was low due to false positive detections. With the ability to search the whole image space using an efficient sampling scheme, WLMCMC produced the lowest error in the *TUD-Campus* and *jogging* sequences. Note that, although WLMCMC searches the whole image space like SemiBoost, its performance is far superior to SemiBoost in almost every sequence. This might be because SemiBoost only does a sliding window based greedy search in a naive manner when

it realizes invisibility of the target, while WLMCMC always searches the whole image space with an advanced and efficient sampling based search strategy.

Moving to sequences containing partial occlusions (Fig. 4.7), SCM produced the lowest error in the *car* sequence, while both SCM and L1-APG had the best performance in the *TUD-Crossing* sequence. SCM uses a sparse based generative model that considers spatial relationships among local patches within an occlusion handling scheme. L1-APG employs a robust minimization model for achieving sparse representation that is also influenced by an explicit occlusion detection mechanism. Thus, both these approaches are quite effective in overcoming partial occlusions; they can identify the occluded part of the target reliably, which is crucial for adaptive appearance models to stay valid under such circumstances. In contrast, MTS-L and MTS-TS use a very simple, generic appearance model (colour histogram), and no explicit occlusion handling mechanism. It might be true that a more complex system complete with more advanced appearance models would obtain a higher overall tracking accuracy. However, employing such a system might complicate attribution of experimental results to the original hypothesis.

Table 4.1(b) reports the percentage of correctly tracked frames based on Pascal score [Santner et al., 2010], and precision at a fixed threshold of 20 pixels. In terms of percentage of correctly tracked frames based on Pascal score, MTS-L maintains the same performance as it showed in table 4.1(a), that it outperforms competing methods in 5 out of 9 sequences. Although the overall performance of MTS-L is unchanged, one of the sequences in the list of 5 is replaced by another when compared to the list in table 4.1(a). This is due to the following: MTS-L has the highest accuracy in the *TUD-Campus* sequence, and WLMCMC performs second best in this sequence, while exactly opposite is observable in the *toy1* sequence. According to the precision at a fixed threshold of 20 pixels, MTS-L loses performance by 1 sequence; it has better precision than the other methods in 4 out of 9 sequences.

Tracking accuracy was also measured when the target was occluded and underwent motion variation at the same time (Table 4.2(a) and Table 4.2(b)). MTS-L produced higher accuracy than the other existing methods in both the sequences. The allocation of particle sets with different spreads from multiple prediction scales lets MTS-L capture an increased search space. VTD performed well in the *squash* sequence with the mean center location error and precision of 20 pixels and 0.78, respectively as it integrates two motion models of different variances, within an MCMC framework, to search a large state space efficiently. WLMCMC produced the second best accuracy on *ball1* as it searches the whole image space using an efficient sampling mechanism to capture abrupt target motion. T_{NCV} and T_{TS}, produced mean center location error of 74 and 81 pixels, respectively in the *ball1* sequence. These trackers use a fixed appearance model

TABLE 4.1: **Tracking accuracy in the presence of occlusion**

(A) Mean centre location error in pixels is given, averaged over all frames of all videos showing occlusions. Each tracker was run five times and the results were averaged. The best results are marked in bold. T denotes the prediction-scales, and N is the number of particles propagated from $t-k$ to t in our proposed method. N is fixed at 20, and N_t is the total number of particles accumulated at time t in our proposed method. The number of particles used in baseline trackers was equal to N_t.

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	T_{NCV}	T_{RW}	T_{TS}	MTS-L	MTS-TS	T	N_t=N×T
ball2	76	71	71	66	78	106	37	91	71	125	17	**16**	32	640
TUD-Camp	186	180	100	186	61	112	**22**	141	119	31	24	**22**	9	180
TUD-Cross	**2**	6	**2**	63	62	5	50	43	75	106	25	21	25	500
PETS 2001	61	63	60	83	114	67	90	43	131	112	25	**21**	32	640
Person	91	80	103	85	177	84	25	90	33	95	10	**8**	20	400
PETS 2009	35	13	81	94	29	10	91	75	37	56	7	**6**	14	280
car	**8**	31	31	47	38	15	28	37	43	87	25	25	20	400
toy1	88	85	111	98	99	107	30	74	134	76	**21**	22	30	600
jogging	110	104	45	70	30	94	**19**	27	24	100	25	24	20	400

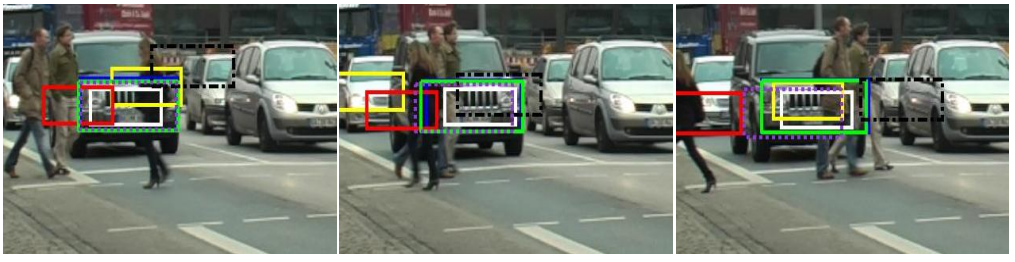
(B) A(B): A - the percentage of correctly tracked frames based on Pascal Score [Santner et al., 2010]; B - Precision at a fixed threshold of 20 pixels. Pascal score is computed by assessing to what extent the tracking template overlaps the ground truth template as a ratio. If the Pascal score is greater than 0.5 in a certain frame, that frame is counted as a correctly tracked frame. Precision is computed by dividing the number of frames, where estimated target location was not beyond the fixed threshold distance of 20 pixels of the ground truth, by the total number of frames in a video sequence. The best results are marked in bold.

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	MTS-L	MTS-TS
ball2	12(0.21)	9(0.17)	7(0.21)	9(0.11)	7(0.13)	9(0.09)	28(0.53)	31(0.8)	**36**(**0.8**)
TUD-Camp	14(0.17)	10(0.14)	19(0.21)	25(0.25)	38(0.34)	27(0.27)	46(**0.61**)	55(0.57)	**57**(0.46)
TUD-Cross	**100**(**1**)	99(0.9)	**100**(**1**)	24(0.23)	41(0.42)	87(1)	25(0.23)	61(0.59)	69(0.65)
PETS 2001	23(0.33)	23(0.27)	22(0.25)	20(0.25)	17(0.2)	16(0.31)	19(0.52)	58(0.65)	**66**(**0.7**)
Person	45(0.46)	44(0.45)	10(0.12)	43(0.45)	20(0.2)	38(0.41)	49(0.86)	79(0.93)	**80**(**0.94**)
PETS 2009	26(0.36)	36(0.7)	21(0.26)	21(0.21)	27(0.45)	65(0.73)	7(0.23)	70(0.97)	**71**(**0.96**)
car	**92**(**0.93**)	62(0.64)	66(0.65)	66(0.65)	55(0.46)	80(0.76)	62(0.52)	71(0.72)	73(0.72)
toy1	18(0.19)	19(0.2)	15(0.15)	16(0.18)	16(0.18)	3(0.09)	**49**(**0.8**)	43(0.8)	38(0.78)
jogging	21(0.22)	22(0.22)	21(0.21)	22(0.22)	**60**(**0.61**)	21(0.21)	42(**0.61**)	20(0.44)	21(0.45)

(A) # 27 (B) # 47 (C) # 78

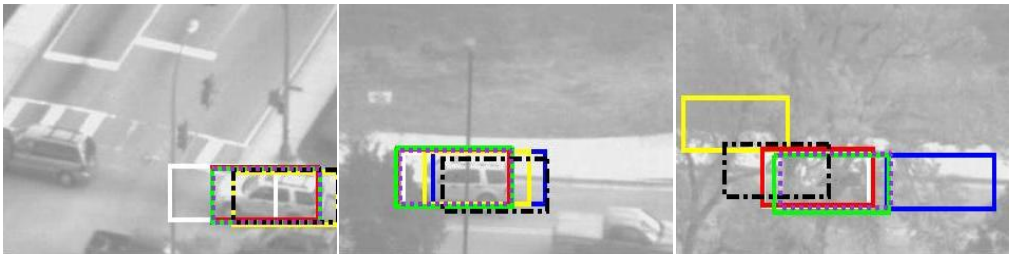
(D) # 27 (E) # 47 (F) # 78

(G) # 93 (H) # 108 (I) # 127

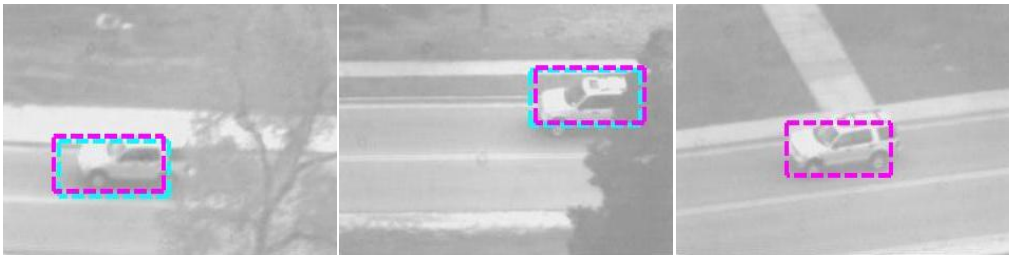


(J) # 93 (K) # 108 (L) # 127

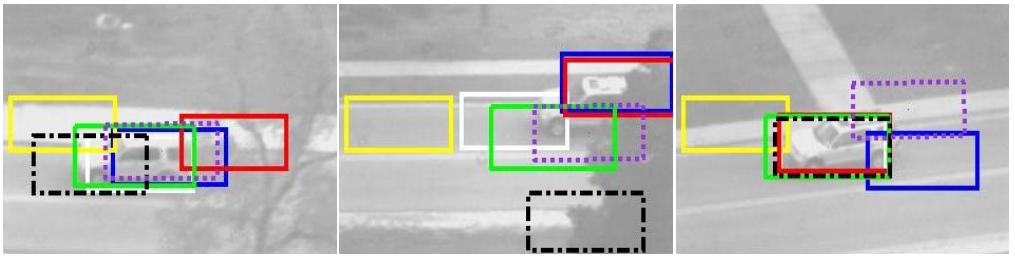
(M) # 49 (N) # 170 (O) # 181



(P) # 49 (Q) # 170 (R) # 181



(S) # 191 (T) # 230 (U) # 315



(V) # 191 (W) # 230 (X) # 315

FIGURE 4.7: **Tracking through multiple partial occlusions.** MTS-TS(magenta), MTS-L(cyan), SCM(green) FragT(white), SemiBoost(yellow), L1-APG(blue), VTD(red), WLMCMC(black), and ASLA(purple).

and maintain a reasonable coverage of the state space, which helped in overcoming shorter occlusions.

Table 4.3 reports accuracy under abrupt motion variations; these variations are due to unexpected motion of the target itself. Along with this rapid target motion, in the *deer* sequence, there are distractors, regions bearing visual similarity to the target, moving alongside the tracked target. ASLA outperforms every other method in the *deer* sequence in all three evaluation metrics, while MTS-L achieves second best accuracy in this sequence in terms of center location error. ASLA uses a local sparse appearance model, which contains both spatial and partial information of the target and makes this method more robust to surrounding clutter than MTS-L, which is based on a holistic representation of the target. FragT tracks 66 percent of the frames successfully (in terms of Pascal score) in the *deer* sequence. It exploits local appearance information, which is in some sense similar to the local sparse appearance model of ASLA. In the *lemming* sequence, both MTS-L and T_{TS} perform equally well, however, MTS-L has higher accuracy. This might be due to the fact that MTS-L generates better location predictions compared to T_{TS} since the former derives motion models from multiple temporal scales whereas the latter depends upon a single-scale motion model. No other tracker except WLMCMC showed comparable accuracy to MTS-L as they drift due to incorrect appearance model updates and cannot recover. In the *boy* sequence, SCM performs better than every other tracker and VTD shows second best result in terms of all three evaluation metrics. MTS-L tracks 91 percent of frames correctly and shows a precision of 0.99, achieving third best accuracy in terms of percentage of correctly tracked frames and precision. Methods based on adaptive appearance models like SCM and VTD show higher performance than the methods based on fixed (non-adaptive) appearance models like MTS-L because the target undergoes out-of-plane rotation in some parts of the sequence.

Note that MTS-TS performs only slightly better than MTS-L. In general, MTS-L produces an accurate approximation of the likely target path. When the target deviates considerably from its predicted location, MTS-TS is a little more accurate; Eq.4.16a spreads the particles more widely to compensate and the combination of the rigid prediction with the flexible estimate (Eq.4.17) reduces the variation of the best state estimate produced by the particles.

In Eq.4.17, the rigid prediction acts as a regularizer to reduce the variance of the flexible estimate [Kristan et al., 2010]. This weighted combination of two different sources is also similar in concept to the fusion mechanism in the Kalman Filter [Kalman, 1960]. In Kalman Filtering, the weights fusing prediction and measurement come from the covariance associated with prediction's and measurement noise [Kristan et al., 2010].

TABLE 4.2: **Accuracy through simultaneous motion variation and occlusion**

(A) Mean centre location error (pixels).

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	T_{NCV}	T_{RW}	T_{TS}	MTS-L	MTS-TS	T	N_t=N×T
squash	40	34	60	20	68	35	22	27	52	41	12	**10**	5	100
ball1	91	96	124	69	66	188	23	74	87	98	15	**14**	14	280

(B) A(B): A - the percentage of correctly tracked frames based on Pascal Score; B - Precision at a fixed threshold of 20 pixels.

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	MTS-L	MTS-TS
squash	60(0.62)	38(0.56)	9(0.11)	68(0.78)	44(0.7)	37(0.5)	50(0.75)	71(0.92)	**75(0.96)**
ball1	6(0.06)	3(0.04)	2(0.05)	19(0.22)	19(0.33)	2(0.02)	35(0.79)	40(0.83)	**41(0.89)**

TABLE 4.3: **Accuracy through abrupt motion variations**

(A) Mean centre location error (pixels).

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	T_{NCV}	T_{RW}	T_{TS}	MTS-L	T	N	N_t=N×T
deer	180	**4**	118	32	76	29	63	142	43	30	21	1	700	700
lemming	63	47	208	77	103	96	17	81	72	15	**13**	1	400	400
boy	**2**	4	32	3	203	17	71	7	57	11	5	1	200	200

(B) A(B): A - the percentage of correctly tracked frames based on Pascal Score; B - Precision at a fixed threshold of 20 pixels.

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	MTS-L
deer	2(0.01)	**100(1)**	9(0.07)	30(0.31)	33(0.35)	66(0.67)	59(0.58)	59(0.67)
lemming	32(0.3)	52(0.52)	2(0.01)	56(0.57)	17(0.15)	50(0.51)	81(0.73)	**86(0.84)**
boy	**98(1)**	89(0.95)	42(0.43)	97(1)	33(0.35)	76(0.79)	34(0.65)	91(0.99)

Whereas in the two-stage model the weights are based on the visual likelihood scores produced by the appearance model.

Fig. A.1 in Appendix A contains precision and success plots for MTS-L, FragT, L1-APG, SemiBoost, VTD, SCM, WLMCMC, and ASLA for all fourteen sequences used. Each precision plot reports precision at a range of thresholds starting from 0 to 50 pixels with an increment of 5 pixels. Similarly, each success plot shows success rate at a range of overlap thresholds beginning from 0 to 1 that is divided in 10 equal intervals.

According to precision plots in Fig. A.1, MTS-L has higher precision than the other existing methods in six out of fourteen sequences within the threshold range of 0 to 20 pixels. In terms of success plots in Fig. A.1, MTS-L achieves higher success rate than the other competing methods in three out of fourteen sequences within the overlap threshold range of 0 to 0.8.

In another experiment the additional cost of the proposed method is quantified in number of particles and is compared with a traditional particle filter with a number of particles equivalent to this additional cost and the total number of propagated particles in the proposed method.

The additional cost of the proposed solution (MTS-L) in terms of number of particles at each time-point is:

$$a = (G \times T) + (G \times V). \tag{4.18}$$

Where G is the cardinality of the set of motion models learnt at each time-point, T denotes the number of prediction-scales, and V denotes the number of variables included in the target state. G is fixed at 4 as mentioned in section 4.5.3. T is determined according to the maximum expected duration (in number of frames) of occlusions in a given sequence (see table 4.1). V is fixed at 3 as there are three elements in the target state, x, y and s.

The first term in Eq. 4.18 denotes the cost of evaluating $G \times T$ state predictions based on visual likelihood score, while the second term represents the cost of learning G motion models.

Given the additional cost of MTS-L according to Eq. 4.18, the number of particles in a traditional Particle Filter is set to $((G \times T) + (G \times V)) + N_t$ for comparison with MTS-L. Where N_t is the total number of particles accumulated at each time-step in MTS-L and it is defined in the caption of table 4.1. The traditional Particle Filter T_{RW} is the colour based Particle Filter from [Pérez et al., 2002], and uses a random-walk motion model.

Fig. 4.8 compares the performance in terms of precision at a fixed threshold of 20 pixels of MTS-L with T_{RW} in five different sequences. The value of T for each sequence is same as mentioned in table 4.1(a). As can be seen, MTS-L achieves improved precision over T_{RW} in all five sequences. This suggests that learning and propagation of tracking information over multiple temporal scales improves tracking accuracy during occlusions and non-constant target motion.

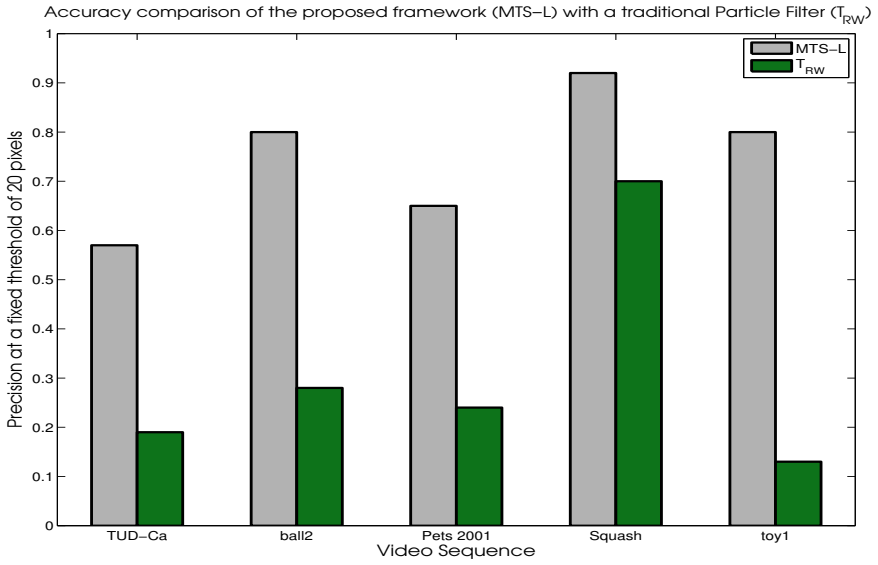


FIGURE 4.8: Performance comparison in terms of precision at a fixed threshold of 20 pixels of MTS-L with a traditional Particle Filter with a number of particles requiring the same amount of resources as MTS-L.

4.5.4.2 Qualitative Evaluation

Tracking is particularly difficult when the time between two consecutive occlusions is small and there is a significant amount of clutter. In *TUD-Campus*, the tracked person suffers two occlusions only 17 frames apart. Fig. 4.9 shows tracking results. MTS-L and WLMCMC recover the target after both the first occlusion (frame # 18), and second occlusion (frame # 40). In contrast, other methods fail due to incorrect appearance model updates, or get distracted by the surrounding clutter. VTD and SCM lock onto the person occluding the target in frame # 18 and keep tracking it for the rest of the sequence. L1-APG, FragT, and ASLA drift and are not able to recover from the occlusions (frame # 18,33, and 40). SemiBoost re-locates the target in frame # 24 and 40, but often detects false positives as shown in frame # 18, 33, and 48.

Occlusions of varying lengths are common in real-world tracking scenarios. In the *person* sequence, a person moves behind several trees, and this sequence is shot with a moving

(A) # 12 (B) # 18 (C) # 24

(D) # 12 (E) # 18 (F) # 24

(G) # 33 (H) # 40 (I) # 48

(J) # 33 (K) # 40 (L) # 48

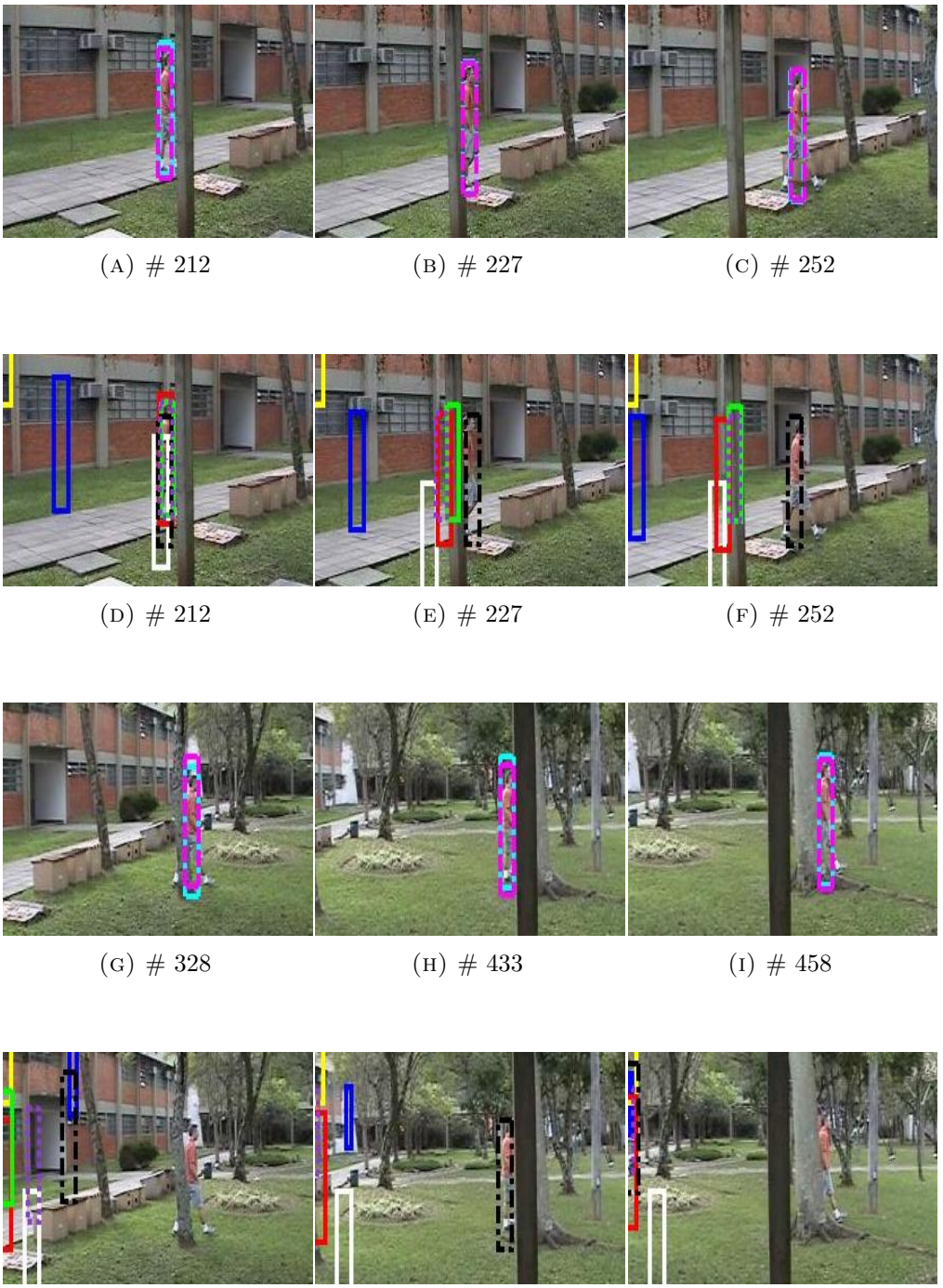
FIGURE 4.9: **Tracking results when the time difference between two consecutive occlusion is small (17 frames).** MTS-TS(magenta), MTS-L(cyan), SCM(green) FragT(white), SemiBoost(yellow), L1-APG(blue), VTD(red), WLM-CMC(black), and ASLA(purple).

camera. As illustrated in Fig. 4.10, L1-APG loses the target earlier than the other methods and drifts into the background (frame # 212), and the other competing methods except WLMCMC fail to re-capture the target after the first occlusion (frame # 227). As WLMCMC searches whole image space, it re-acquires target after occlusions (frame # 227, and frame # 433), and therefore performs better than the other competing methods. However, it is also distracted by the surrounding clutter (frame # 328, and frame # 458). In comparison to the competing methods, MTS-L accurately re-captures the target after each occlusion, and is less prone to distractors (frame # 227, 328, and 458).

Video surveillance data often requires tracking through partial and/or full occlusions. In the *PETS 2001 Dataset 1* sequence (Fig. 4.11) the target (car) first stays partially occluded for 25 frames, and is then completely occluded for 31 frames by a tree. Moreover, the target shrinks significantly after it re-appears as it is moving away from camera. Fig. 4.11 shows tracking results. All the methods track the target before it hides behind the tree (frame # 46), but only WLMCMC and MTS-L recover the target when it re-appears (frame # 93, and 178) after being fully occluded.

Fig. 4.12 shows another example of occlusion caused by a stationary object while tracking in an outdoor environment. Since the target is jogging, both its appearance and the dynamics of background are changing. From Fig. 4.12 it can be seen that all the methods except WLMCMC, SemiBoost, and MTS-L fail to recover the target after occlusion (frame # 86). Although all three trackers, WLMCMC, SemiBoost, and MTS-L, track the target till the end of sequence, SemiBoost shows the best accuracy (frame # 171, 259, and 307). SemiBoost can cope with the changing appearance as it adapts to these variations with an online semisupervised boosting algorithm, while the other two cannot since they use a fixed appearance model.

The ability of MTS-L to cope with simultaneous occlusion and non-constant target motion was tested by making two challenging sequences: *squash* and *ball1*. In these sequences, the target displays abrupt motion, accelerates, decelerates, changes direction suddenly, and is completely occluded multiple times. Fig. 4.13(a-l) illustrates tracking results on the *squash* sequence. MTS-L provides more accurate tracking in the *squash* sequence than any other method. This is because the combination of learning models over multiple model-scales and applying them over multiple prediction-scales creates a rich and improved prior distribution that allows the algorithm to cover a relatively large search space. VTD also performs quite well on this sequence (frame # 148, 237, and 243) because its combination of basic appearance and motion models allows capture of appearance variations, and provides robustness against shorter occlusions and rapid motion variation.



(A) # 212 (B) # 227 (C) # 252

(D) # 212 (E) # 227 (F) # 252

(G) # 328 (H) # 433 (I) # 458

(J) # 328 (K) # 433 (L) # 458

FIGURE 4.10: **Tracking results with occlusions of different lengths in an outdoor environment.** MTS-TS(magenta), MTS-L(cyan), SCM(green) FragT(white), SemiBoost(yellow), L1-APG(blue), VTD(red), WLMCMC(black), and ASLA(purple).

(A) # 16 (B) # 27 (C) # 46

(D) # 16 (E) # 27 (F) # 46

(G) # 63 (H) # 93 (I) # 178

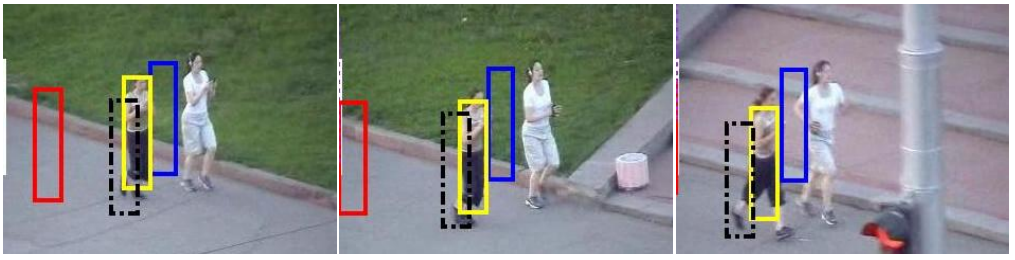
(J) # 63 (K) # 93 (L) # 178

FIGURE 4.11: **Tracking results in a surveillance environment.** MTS-TS(magenta), MTS-L(cyan), SCM(green) FragT(white), SemiBoost(yellow), L1-APG(blue), VTD(red), WLMCMC(black), and ASLA(purple).

(A) # 44 (B) # 63 (C) # 86

(D) # 44 (E) # 63 (F) # 86

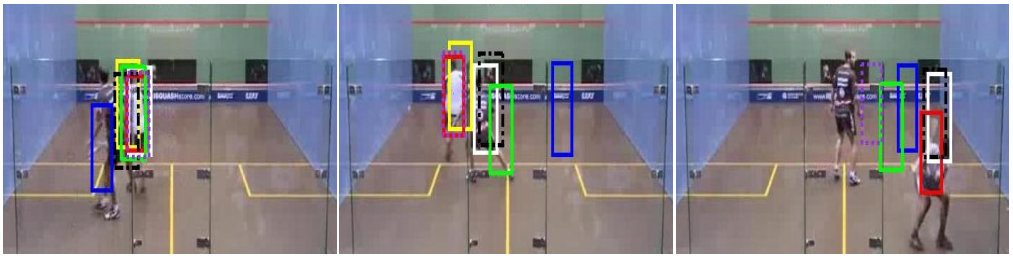
(G) # 171 (H) # 259 (I) # 307



(J) # 171 (K) # 259 (L) # 307

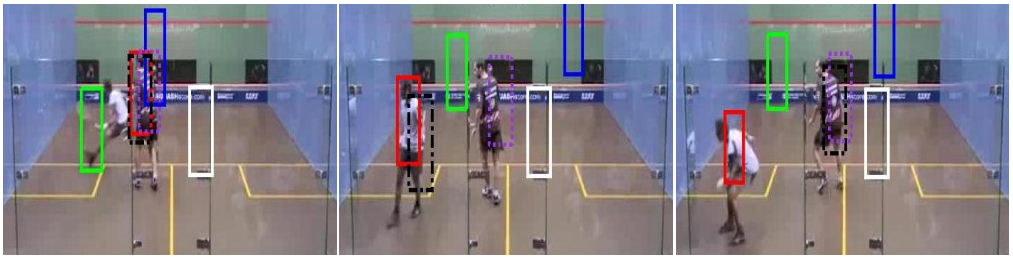
FIGURE 4.12: **Another example of occlusion in an outdoor environment.** MTS-TS(magenta), MTS-L(cyan), SCM(green) FragT(white), SemiBoost(yellow), L1-APG(blue), VTD(red), WLMCMC(black), and ASLA(purple).

(A) # 68 (B) # 125 (C) # 148

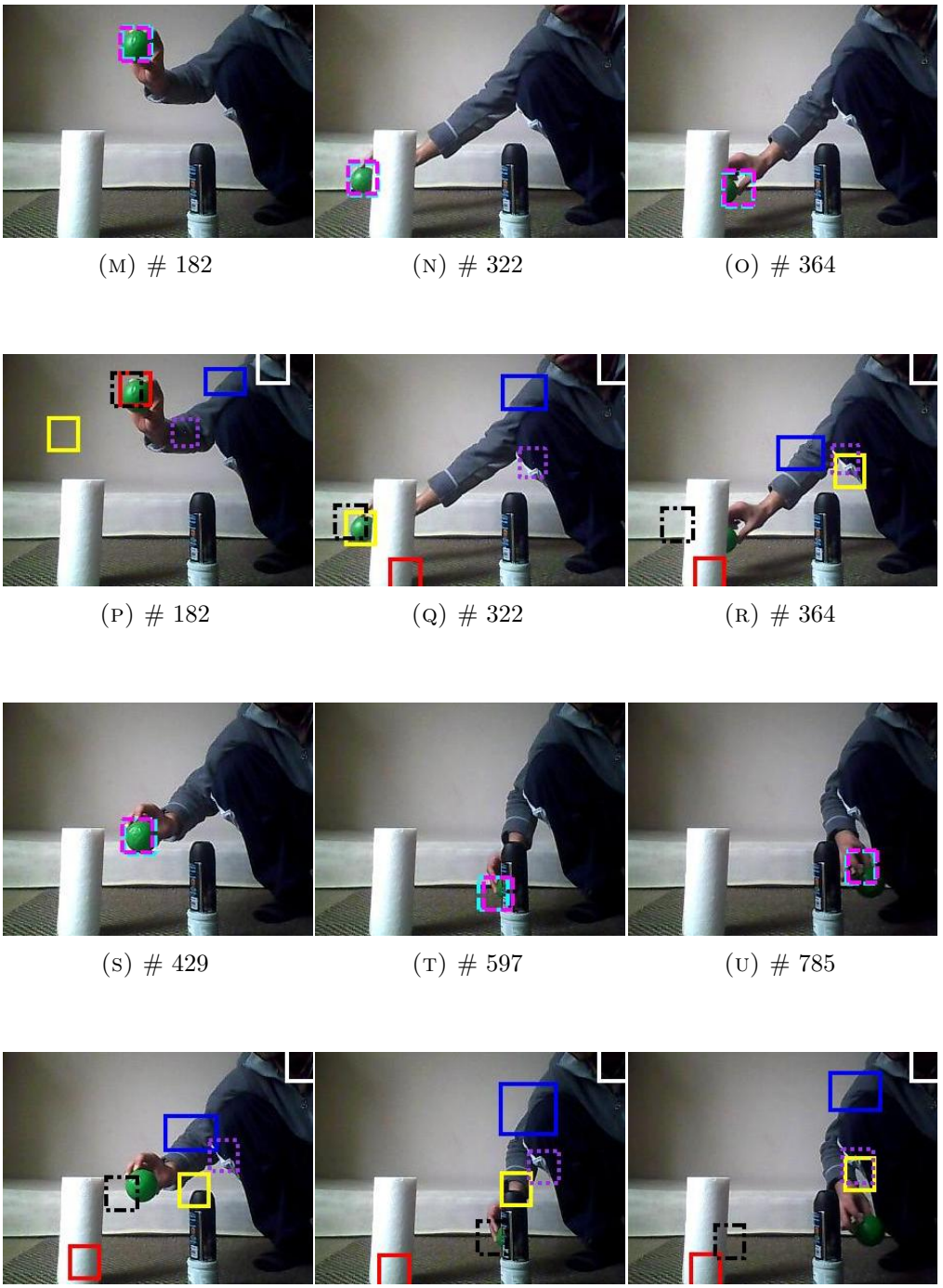


(D) # 68 (E) # 125 (F) # 148

(G) # 194 (H) # 237 (I) # 243



(J) # 194 (K) # 237 (L) # 243



(M) # 182 (N) # 322 (O) # 364

(P) # 182 (Q) # 322 (R) # 364

(S) # 429 (T) # 597 (U) # 785

(V) # 429 (W) # 597 (X) # 785

FIGURE 4.13: **Tracking results in case of abrupt motion variations and frequent occlusions.** MTS-TS(magenta), MTS-L(cyan), SCM(green) FragT(white), SemiBoost(yellow), L1-APG(blue), VTD(red), WLMCMC(black), and ASLA(purple).

The *ball1* sequence is more challenging because the target is occluded partially and completely for variable time periods. Six samples of the tracking results are shown in Fig. 4.13(m-x), with frame numbers 182, 322, 364, 429, 597, and 785. None of the trackers except MTS-L and WLMCMC is able to track the target throughout, and MTS-L shows the best performance in this sequence. This is because the cluster of particles propagated from several previous time-points through models generated over different model-scales lets MTS-L capture correct modes of the complex target distribution, to be estimated in these conditions. WLMCMC shows robustness against occlusions and abrupt motion variations to some extent, but when the target changes appearance it can be distracted by an object (in the scene) whose appearance is similar to the fixed target model as it searches the whole image space (frame # 364, and 785).

(A) # 31 (B) # 32 (C) # 31 (D) # 32

(E) # 341 (F) # 365 (G) # 341 (H) # 365

FIGURE 4.14: A comparison of tracking results between MTS-L(cyan), T_{NCV}(blue), T_{RW}(magenta), and T_{TS}(white) during rapid motion variation, and occlusion.

Fig. 4.14 demonstrates a comparison between MTS-L, T_{NCV}, T_{RW}, and T_{TS} in two cases: a target displaying rapid motion, and are completely occluded. Frame # 31, and 32 show rapid movement of the target towards the left. The competing trackers lose the target, while MTS-L tracks it. Efficient search with particles of different spreads around the likely paths of the target, which are defined by motion models belonging to several previous time-points and generated over different model-scales, lets MTS-L cover the increased search space generated by this rapid motion variation. Frame # 341, and 365 display the target before and after full occlusion, respectively. T_{RW}, T_{TS} fail

to re-capture the target after occlusion in Frame # 364. In contrast, the presence of at least one valid set of particles propagated from some previous time-step by a motion model derived from an appropriate model-scale allows MTS-L to reliably recover the target.

4.5.5 Analysis of the Proposed Framework

4.5.5.1 Without Multiple Prediction-Scales

The proposed tracker was tested without employing multiple prediction-scales. For this purpose, MTSWPS-L was designed. MTSWPS-L stands for MTS-L without employing multiple prediction-scales. In MTSWPS-L, the target state is predicted only 1 frame ahead i.e. $T = 1$. For evaluation, at first, the number of particles in MTSWPS-L was kept equal to N_t and the process noise σ_{xy} was same as used for MTS-L between two consecutive time-steps. To analyze further, later, both the number of particles N_t and the process noise σ_{xy} were doubled and tripled. Fig. 4.15 reveals the performance of the proposed method with and without multiple prediction-scales in five video sequences involving occlusions. As can be seen, MTSWPS-L has poor performance compared to MTS-L in all 5 sequences even after increasing the sampling effort and the process noise by three times of the original. Therefore, it can be said that operation over multiple prediction-scales allows the proposed method to reliably handle occlusions in a principled way.

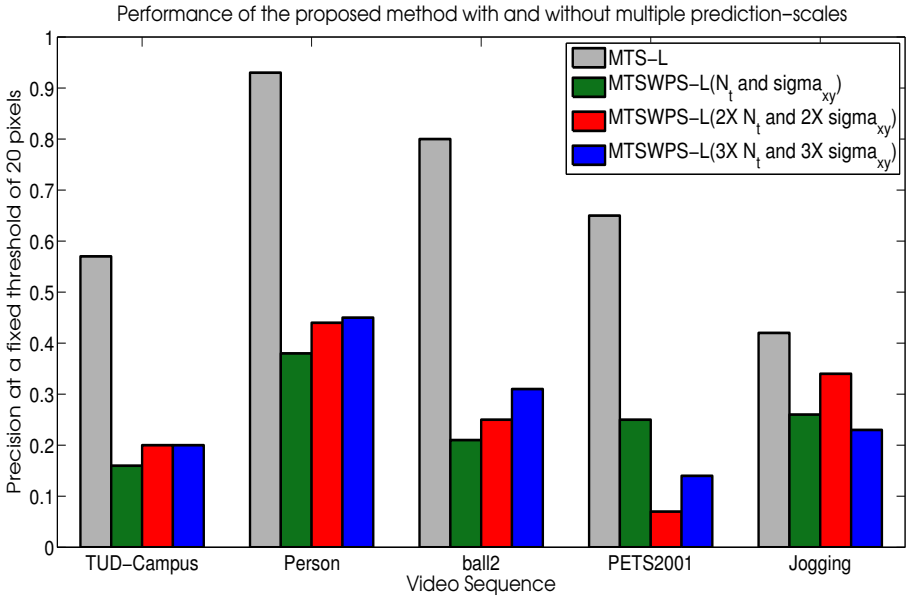


FIGURE 4.15: Performance of the proposed framework with and without multiple prediction-scales.

4.5.5.2 Without Multiple Model-Scales

The proposed tracker was also analyzed without learning over multiple model-scales. MTSWMS-MX denotes the proposed method in which a linear motion model is learned over model-scale X only. As a result, there is no need to select models from each of the previous time-steps at the current time-step since only 1 model is learned over a single model-scale. As can be seen in Fig. 4.16, MTS-L has superior performance over MTSWMS-M2, MTSWMS-M3, MTSWMS-M4, and MTSWMS-M5 in all nine sequences. This suggests that by constructing motion models over multiple model-scales MTS-L maintains a richer description of the target's path than is possible with a single scale model. Furthermore, this diverse set of models produces motion priors that ultimately develop into a rich prior distribution required for reliable recovery of tracking after occlusions.

Fig. 4.16 shows results from three different kinds of sequences. The first four sequences contain abrupt motion variations, next three involve occlusions and abrupt motion variations, and the last two comprise of occlusions and smooth motion variations. Trackers based on shorter scale models generally outperform trackers based on longer scale models in the *lemming*, *toy2*, *boy*, and *deer* sequences. However, there is no such ranking pattern in other sequences. This might be because in these sequences the target state is predicted over multiple prediction-scales and the search is performed with particle sets having different spreads.

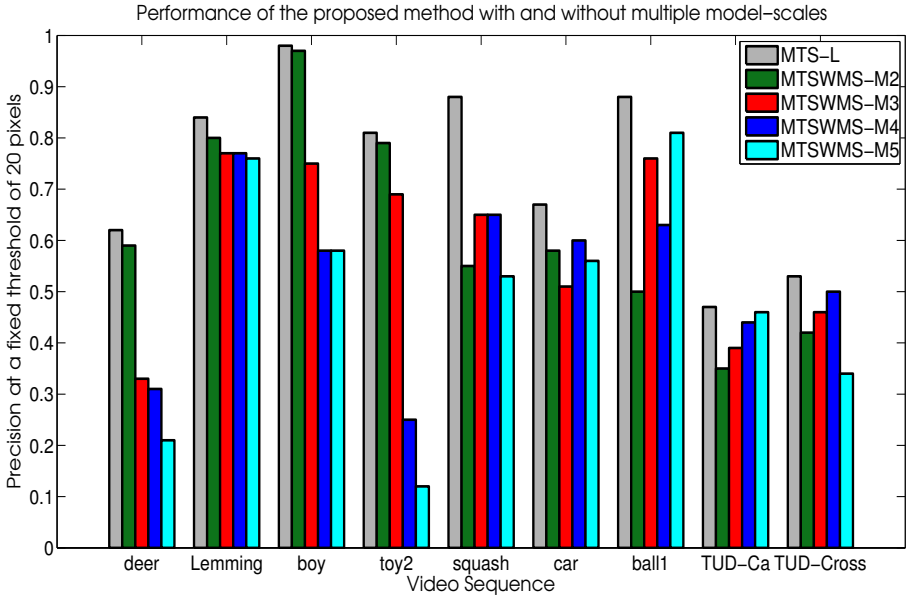


FIGURE 4.16: Performance of the proposed framework with and without multiple model-scales. MTSWMS-M2, MTSWMS-M3, MTSWMS-M4, and MTSWMS-M5 denote the proposed method in which a linear motion model is learned only over model-scale 2,3,4, and 5, respectively.

4.5.5.3 Varying the Degrees of (Polynomial) Motion Models

The performance of the proposed tracker was also tested by varying the degrees of the polynomial motion models used. MTS-Z, MTS-L, and MTS-Q denote the proposed method applied over a zero-order, first-order, and second-order polynomial motion models, respectively. The model-scales used for MTS-L were as mentioned earlier. For MTS-Q they ranged from 6 to 9 frames, because more data points (estimated states) are required in case of a second-order polynomial to avoid the risk of overfitting. In MTS-Z, the state $\mathbf{X}_t$ of the target at time t is predicted by adding (Gaussian) noise to the state at time $t-1$. The noise levels in MTS-Z were adjusted to produce the best tracking performance.

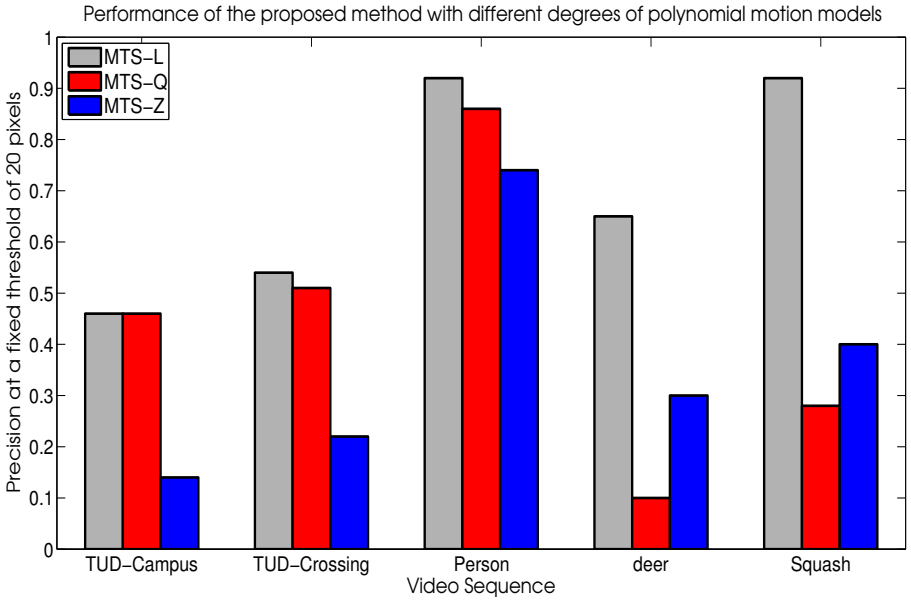


FIGURE 4.17: Performance of the proposed framework with different degrees of polynomial motion models.

Fig. 4.17 reveals the performance of MTS-Z, MTS-L, and MTS-Q in five different sequences. As the state predictions made by the second-order polynomial are often inaccurate during rapidly varying target motion, MTS-Q shows worst performance in the *deer* and *Squash* sequences. However, MTS-Q outperforms MTS-Z in situations where target motion is relatively smooth and occlusions are longer. This is because state predictions made by just adding noise to the single (recently estimated) state can only account for the radical acceleration of the target, and are not accurate enough when the target is moving in a straight line. In contrast, a first-order polynomial motion model (linear motion model), learned over different model-scales, can capture a range of variation in the target motion by generating accurate state predictions both during

smooth and non-constant motion. So, MTS-L shows better performance over MTS-Z and MTS-Q in each sequence.

4.5.5.4 Fixed Number of Prediction-Scales

The proposed framework was evaluated by keeping the number of prediction-scales (the T parameter) fixed in all those videos that involve occlusion or motion variations and occlusions. The aim is to observe the sensitivity in terms of performance of the proposed framework to the T parameter. For evaluation, T is fixed to 32 (frames) for all the videos. T is set to 32 as the expected maximum duration of occlusion in each video is less than 32 frames. Please note that the maximum length of occlusion in most of the videos is not the same, and it shows considerable variation. N, the number of particles propagated from time $t-k$ to time t, was the same (fixed to 20) as mentioned in tables 4.1(a) and 4.2(a). Fig. 4.18 shows the performance of the proposed framework with and without fixed number of prediction-scales in eleven video sequences. It can be seen that the performance of the proposed framework does not deteriorate much even when fixing T to a certain number.

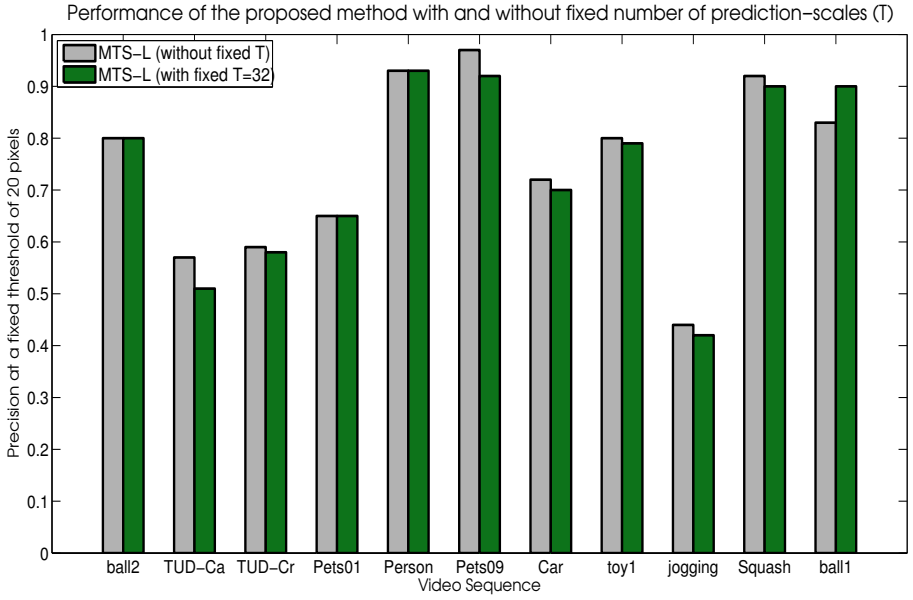


FIGURE 4.18: Performance of the proposed framework with and without fixed number of prediction-scales (T parameter). For MTS-L(fixed T), the value of T is set to 32 for all video sequences, and for MTS-L(without fixed T), the value of T for each video sequence is same as shown in tables 4.1(a), and 4.2(a).

The notable decrease in accuracy for MTS-L(fixed T) in the *Pets'09* and *TUD-Campus* sequences when compared to MTS-L(without fixed T) is due to the following. When the proposed tracker predicts further into the future i.e. T becomes larger, the likelihood

of meeting false-positives increases as the number of inaccurate predictions and the spread of particles around some inaccurate prediction become higher. Since the actual maximum duration of occlusion is half of (or even smaller than) the fixed value of T in both these sequences, MTS-L(fixed T) gets distracted by the surrounding clutter more often than MTS-L(without fixed T) during partial or full occlusions.

4.5.5.5 Potential Drawbacks

The experimental results reveal the robust performance of the proposed method during occlusions. However, the proposed method is susceptible to failure when faced with long duration occlusions. At very large prediction scales, the spread of the particles would be so large that they might miss the target when it becomes visible after staying occluded for a very long time.

Another interesting case is the presence of a visually similar object close to the tracked target when it re-appears after occlusion. If the estimated states during the period of occlusion are not nearer to the true target states, then the corresponding learned motion models can make inaccurate state predictions. As a result, the proposed tracker can be distracted by a visually similar object. This is due to the fact that the proposed method uses a naive model-selection criterion, which is based on the likelihood of a very simple appearance model and selects one motion model from each of the T previous time-points. Furthermore, the proposed tracker just allocates a fixed number of particles around state predictions without taking into account their location in the target distribution.

4.6 Conclusion

This chapter proposes a tracking framework capable of handling occlusion and abrupt motion variation by exploiting multiple temporal scales. This is achieved by learning motion models over multiple model-scales and applying those over multiple-prediction-scales. A simple strategy, based on the visual likelihood score of a fixed appearance model, is used to select one motion model from each of the previous time-steps. The search around the predicted states is accomplished by propagating particles from the several previous time-points. After combining sufficient sets of particles at each time-point, it is assumed that at least one set will be valid and allow recovery from occlusion and robustness to non-constant motion. It is important to note that these particle sets are not, however, simply spread widely across the image: each represents an estimation of the posterior probability from some previous time-point, predicted by a motion model generated over an appropriate model-scale.

The proposed framework was compared to the other methods that either are capable of handling occlusion and abrupt motion variation or have shown good performance on the CVPR'13 [Wu et al., 2013] benchmark. Quantitative and qualitative evaluation show that the proposed framework has superior performance over the competing methods on the both publicly available benchmarks and some new video sequences.

Experiments were conducted to analyze the performance of the proposed tracker by including and excluding multiple temporal scales, varying degrees of polynomial motion models, and keeping the number of prediction-scales fixed. Results show that operation over multiple prediction-scales allows reliable recovery of a target after occlusion. They reveal that the construction of models over multiple model-scales allows handling of variable motion better than any single scale model and their application over multiple prediction-scales supports occlusion handling. In addition, a linear motion model might be more suitable than zero or second order (polynomial) motion model in utilizing the true potential of learning over multiple model-scales. Also, the proposed tracker shows good performance even after keeping the number of prediction-scales the same in nearly all the sequences.

Although the proposed tracker has shown good performance in solving the two outstanding problems in visual tracking, its accuracy may be further improved by exploring its two major components: search method, and model selection criterion. In the context of the proposed framework, the search method seeks the optimal target state around a number of state predictions made by selected motion models, and the model selection criterion chooses some subset of motion models from the space of available motion models at each time. The role of search method is important because it is required to locate and refine possible local maxima in a complex distribution in an efficient manner. The current formulation simply allocates fixed number of particles with different spreads around state (selected) predictions without considering their locations in the target distribution. The role of model selection criterion is even more crucial since it should choose models whose predictions are close to the possible mode of the target distribution, and yet small in number. The current selection strategy picks one motion model from each of the previous time-points, based on a likelihood score, without considering that predictions from some previous time-point might be inaccurate and should be ignored completely. The next chapter generalizes a search method to find the best possible target state around multiple competing hypotheses (state predictions), and studies the problem of model-selection to develop a new selection strategy.

Chapter 5

A Generalized Search Method, and a New Model Selection Criterion

In the preceding chapter, the power of multiple temporal scales of motion model generation and application to deal with visual tracking problems was investigated. The proposed tracker demonstrated promising performance in overcoming occlusions and abrupt motion variations. However, it is important to note that this tracker uses a very simple search method, and a naive motion model-selection strategy. As a result, it might not always be able to reach the true solution, can be wasteful in terms of sampling efficiency, and vulnerable to distractors.

As a first step towards solving the aforementioned problems, and to gain a further insight into the proposed tracking framework (MTS), this chapter explores the two main components of MTS: the search method, and the model selection criterion. In particular, a search method is generalized to estimate the best target state around multiple competing hypotheses generated in this framework, and the problem of model selection is studied to devise a new selection criterion. Here, the search method has to locate local maxima around the multiple competing hypotheses (state predictions) in an efficient manner, and the model selection criterion needs to pick a small number of hypotheses that are close to the likely modes of the target distribution.

Section 5.1 generalizes a search method, originally proposed to search a fixed grid of equal sized cells, to cells of variable size and location formed around the predictions generated by motion models in MTS. The modified framework, which is based on this new search method, is compared to the original instantiation of MTS, which utilizes

particle filters, and performance is reported in Section 5.1.4. Section 5.2 first defines the model selection problem in the context of MTS, then formulates a new (motion) model selection criterion, and finally employs the generalized search method to estimate target state. In Section 5.2.3, the performance of MTS with this new selection strategy is compared to MTS with the existing model set reduction method (proposed in Chapter 4).

5.1 A Generalized Search Method for Multiple Competing Hypotheses in Visual Tracking

Generally speaking, a tracking algorithm either estimates the correspondence and target detection jointly, or performs detection in each frame followed by a data association stage [Yilmaz et al., 2006]. Here, the methods belonging to the first category are under discussion that use a matching function and a search strategy to jointly estimate the target track and target region.

The matching function weighs how well a certain hypothesis matches the target model, while the search strategy finds the optimal hypothesis through maximising or minimising an objective function, which itself is a function of the matching function. In this section, a search method is generalized to obtain the best hypothesis from multiple competing hypotheses arbitrarily positioned in the search space. These hypotheses are state predictions generated by motion models in MTS (the tracking framework proposed in Chapter 4).

Two different search strategies are commonly used by visual trackers: gradient descent and stochastic methods. Gradient descent methods [Comaniciu et al., 2003],[Yang et al., 2005] remain popular due to their fast convergence rate and low computational cost, but can become trapped in local modes of the filtering distribution due to e.g. background clutter or rapid motion of a target. Stochastic methods such as particle filters (PF) [Pérez et al., 2002],[Isard and Blake, 1998a],[Li et al., 2008] have enjoyed much success in tracking, as they can handle non-Gaussianity and multi-modality of a target distribution. PF is computationally impractical for the high dimensional spaces typically found in multi-object tracking. In the recent past, many methods [Cappé et al., 2007] have been proposed to reduce the computational expense and improve the efficiency of PF. Among them, Markov Chain Monte Carlo (MCMC) methods gained popularity as efficient search methods [Khan et al., 2005],[Smith et al., 2005]. While simulating a target distribution with deep local maxima these methods can, however, become stuck at a local maximum, leading to an inaccurate Bayesian inference. This is also known as the local trap problem.

Adaptive MCMC algorithms [Roberts and Rosenthal, 2009] provide an automatic way of tuning the proposal variance to maintain a certain acceptance rate of the sampler, and thus can better mix between different modes of a target distribution. However, they do not provide a systematic way of escaping local maxima. Kwon and Lee [Kwon and Lee, 2008], combined the Wang-Landau Monte Carlo method with the MCMC method to escape local maxima in a complex target distribution, searching in a regular grid that divides the image space in a number of equally sized cells. Towards a similar goal, a Stochastic Approximation Monte Carlo (SAMC) based tracking algorithm was proposed by [Zhou et al., 2012] to search for the optimal target state in a regular grid.

An important ingredient of visual tracking is the motion model. Visual tracking frameworks have typically made use of a single general purpose motion model like Random Walk (RW) or Nearly Constant Velocity (NCV) as it is hard to come up with an accurate motion model for various tracking scenarios. Although they are general, they can result in poor tracking accuracy in situations where a target can display complex motion variations.

To capture different ways a target can move, some attention has been given to the notion of multiple motion models. [Isard and Blake, 1998b] learned a few distinct motion models, and a fixed finite state machine relating transitions among them from ground truth data. Instead of learning from some offline data, Kwon and Lee [Kwon and Lee, 2011] sampled motion models from a pool, generated by utilizing the recent sampling history, to enhance the accuracy and efficiency of the state sampling process. Kristan et al. [Kristan et al., 2010] designed a two-stage dynamic model to improve the accuracy and efficiency of the bootstrap particle filter in handling various target motions. To handle complex target motion and occlusions, the tracker proposed in the previous chapter combines motion models learnt and applied over multiple temporal scales with an extension of the bootstrap particle filter. For a detailed review of the prior work on multiple motion models in visual tracking, please refer to Chapter 2 of this thesis.

Trackers employing multiple motion models such as the proposed framework (MTS) produce multiple competing hypotheses or state predictions as illustrated in the first row of Fig. 5.1. The question then becomes how to search for the optimal target state given these predictions. Here, the search is modelled by assigning each state prediction a certain area in state space, which is called a cell. The size of this cell is proportional to the uncertainty attached with its corresponding prediction, and its position in space depends on an estimator such as a motion model. The second row of Fig. 5.1 describes this problem in a 3D state space. It is believed that the occurrence of the problem of how to search around multiple competing hypotheses generated by multi-scale motion

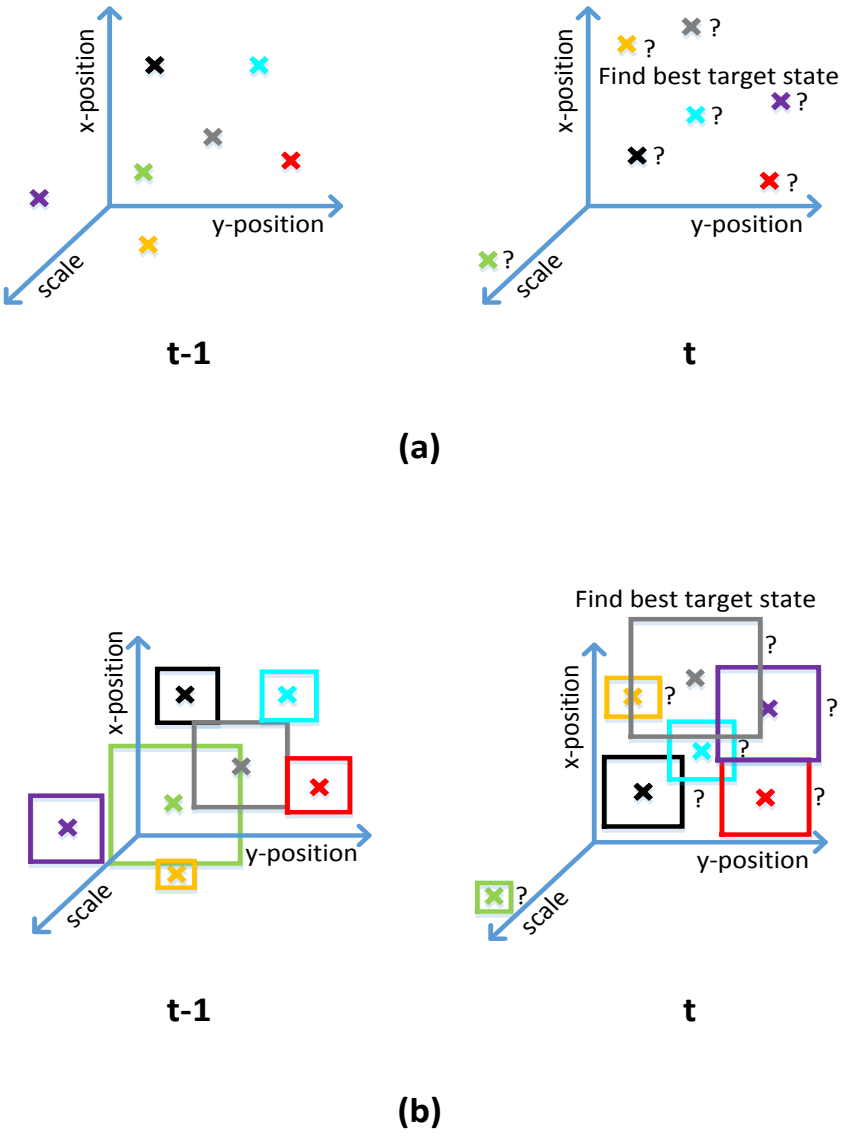


FIGURE 5.1: **Graphical Illustration of the search problem in the proposed framework**. The proposed framework generates multiple competing hypotheses or state predictions. The aim is to find the best target state from these predictions. Fig. 5.1(a) shows multiple state predictions in 3D state space at time t-1 and time t. We propose to model our search by allocating each state prediction a certain area, which we call a cell, in state space. The size of this cell is proportional to the confidence of its corresponding prediction. Fig. 5.1(b) shows cells of variable size in 3D state space at time t-1 and time t, where each cell is formed around a certain state prediction. The question of how to search for the optimal target state in these variable sized cells raises the possibility of a broader range of search strategies that can be introduced.

models in visual tracking invites the possibility of a wider range of search strategies for finding the optimal target state.

To search for the best target state in this scenario, a sampling based search method is generalized, which integrates the Wang-Landau Monte Carlo method and MCMC method (WLMCMC sampling) [Kwon and Lee, 2008] although other density exploration methods such as [Zhou et al., 2012] can also be generalized. WLMCMC operates on a regular grid of equal sized cells. Here, it is generalized to arbitrarily placed cells of variable size. In [Kwon and Lee, 2008], the Wang-Landau method estimates the Density of States (DOS) term, which denotes the extent to which cells have been explored, and this term is used to generate moves to cells that have not been explored enough. This allows discovery of local maxima in specific cells, while jumping between them. The likelihood term in MCMC causes this method to spend more time in cells that contain highly probable target states. With this term, the method expends more samples around the current local maximum, which has already been well explored.

5.1.1 Bayesian Tracking Formulation

The aim is to find the best state of the target at time t given observations up to time t. The state at time t is given by $\mathbf{X}_t = \{X_t^x, X_t^y, X_t^s\}$, where X_t^x, X_t^y, and X_t^s represent the x,y location and scale of the target, respectively. The posterior distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$, given the state $\mathbf{X}_t$ at time t and observations $\mathbf{Y}_{1:t}$ up to t, is estimated using the Bayesian formulation

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto p(\mathbf{Y}_t|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) d\mathbf{X}_{t-1}, \tag{5.1}$$

where $p(\mathbf{Y}_t|\mathbf{X}_t)$ denotes the observation model, and $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is a motion model. Now the best state of the target $\hat{\mathbf{X}}_t$ is obtained using Maximum a Posteriori (MAP) estimation over the N_t particles which approximates the posterior distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$:

$$\hat{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t^{(i)}} p(\mathbf{X}_t^{(i)}|\mathbf{Y}_{1:t}) \; for \; i = 1, ..., N_t, \tag{5.2}$$

where $\mathbf{X}_t^{(i)}$ is the i_{th} particle.

The analytical solution to Eq.5.1 is intractable in practice if the filtering distribution is non-Gaussian. Conventional tracking frameworks typically use a single motion model such as Random Walk and a fixed sampling based search strategy like Particle Filter to approximate $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ [Dellaert et al., 1999, Jia et al., 2012, Koller-Meier and Ade, 2001, Mei et al., 2011, Ross et al., 2008]. Here, the framework being discussed builds motion models over multiple temporal scales and applies them multiple frames ahead,

and thus, generate multiple competing hypotheses or state predictions. For such cases, PF [Pérez et al., 2002] and Metropolis Hastings (MH) [Hastings, 1970] are infeasible, and a broad range of search strategies needs to be explored. PF and MH depend upon a single motion model and operate on a first-order Markov assumption [Dellaert et al., 1999, Khan et al., 2005, Kwon and Lee, 2010, Mei and Ling, 2009, Ross et al., 2008].

5.1.2 A Multiple Temporal Scale Framework

This section briefly overviews the proposed framework (MTS), which was proposed in Chapter 4, to reveal the source of hypotheses around which the best target state is to be sought.

MTS learns motion models at different model-scales, and applies those models at multiple prediction-scales. The application of learned models at multiple prediction-scales generates multiple competing hypotheses or state predictions at each time point. To search for the best target state, MTS extends PF [Arulampalam et al., 2002], in which a fixed particle set with a certain spread is allocated around each state prediction.

To capture possibly complex motion patterns, MTS learns linear motion models at different model-scales. A linear motion model is represented by $\mathbf{M}$. $\mathbf{M}$ is learned at a given model-scale separately for the x-location, y-location, and scale s of the target's state.

Let $Z^m = \{\hat{x_n}\}_{n=t-m+1}^{n=t}$ represent a sequence of recently estimated x-locations of target states at model-scale m. Given M learned at model-scale m at time t, for x-location of target state it is written as:

$$\tilde{x}_n = \phi_{\hat{x}_t}^m + \tau_{\hat{x}_t}^m n, \tag{5.3}$$

where τ is the slope, ϕ the intercept, and $\hat{x}_t$ denotes that the model parameters have been learnt using a sequence of recently estimated x-components of target states whose last member is $\hat{x}_t$. Model parameters are learnt using Weighted Least Squares (WLS) method as described in section 4.3.2 of chapter 4. Eq. 5.3 is same as Eq. 4.13 and Eq. 3.1 and it is repeated here to make the chapter self-contained.

These models are learned at each time t, and a set of these models is represented by $\mathbf{M}_t^{j=1,\ldots,|\mathbf{M}_t|}$, where $|.|$ is the cardinality of the set. Each model predicts target state $l(\tilde{x}, \tilde{y}, \tilde{s})$ at T prediction-scales.

Suppose there are T sets of motion models available at time t, one from each of T previous time-steps. Each set of models at time t is represented by its corresponding set of predictions. The most suitable motion model from each set is selected as follows.

Let us denote $G = |\mathbf{M}_t|$, and let $\mathbf{l}_t^k = \{l_t^{j,k} | j = 1, ..., G\}$ represent a set of states predicted by G motion models learnt at time $t-k$, where $l_t^{j,k}$ denotes the predicted state by j_{th} motion model learned at k_{th} previous time-step. As $k = 1, ..., T$, there are T sets of predicted states at time t. Now the most suitable motion model $\mathbf{R}_t^k$ is selected from each set using the following criterion:

$$\hat{l}_t^k = \arg\max_{l_t^{j,k}} p(\mathbf{Y}_t | l_t^{j,k}) \tag{5.4}$$

where $\hat{l}_t^k$ is the most suitable state prediction from the set $\mathbf{l}_t^k$, and $p(\mathbf{Y}_t | l_t^{j,k})$ measures the visual likelihood at the predicted state $l_t^{j,k}$. In other words, $\hat{l}_t^k$ is the most suitable state prediction of the most suitable motion model $\mathbf{R}_t^k$. After this selection process, the T sets of motion models are reduced to T individual models.

There exist T most suitable state predictions at time t, and it is required to search for the best possible target state around these predictions (see the top row of Fig. 5.2).

The search is modelled by allocating each state prediction $\hat{l}_t^k$ a certain area in the state space, which is called a cell. The size of this cell is equivalent to the uncertainty attached to its corresponding hypothesis (see the bottom row of Fig. 5.2). For instance, the size of the cell around $\hat{l}_t^k$ will be $2 \times \sigma_x \times k$ pixels, and $2 \times \sigma_y \times k$ pixels, respectively. Along the third dimension, scale, the uncertainty would be $\sigma_s \times k$ around the predicted state $\hat{l}_t^k$. Here σ_x, σ_y, and σ_s are the standard deviations of a zero-mean Gaussian noise acting on translation x and y, and scale s between two consecutive time-steps, respectively.

Note that the cell size corresponding to the prediction generated from time $t-1$ is smaller compared to the cell size corresponding to the prediction originated from time $t-2$. It is assumed here that uncertainty in the (motion) information, which is coming from several previous time-steps to time t, increases with temporal distance from the current time-step t.

Given T cells of variable size, which might overlap, the aim is to search for the best target state $\hat{\mathbf{X}}_t$ in these cells. An intuitive and pragmatic approach would be to visit all the cells to some extent, and spend more time in those where there are highly probable target states. Moreover, these cells can be far apart in space. Sampling based search methods such as the MH algorithm [Hastings, 1970], used conventionally in tracking frameworks, cannot be used here. While searching a large area with large proposal variance, the MH algorithm has the tendency to become stuck in local maxima. With a smaller variance, it would require many samples to search a large area, and again get trapped in a local maximum if there are deep valleys between the different modes of the target distribution. WLMCMC sampling is generalized in the next section to solve these problems.

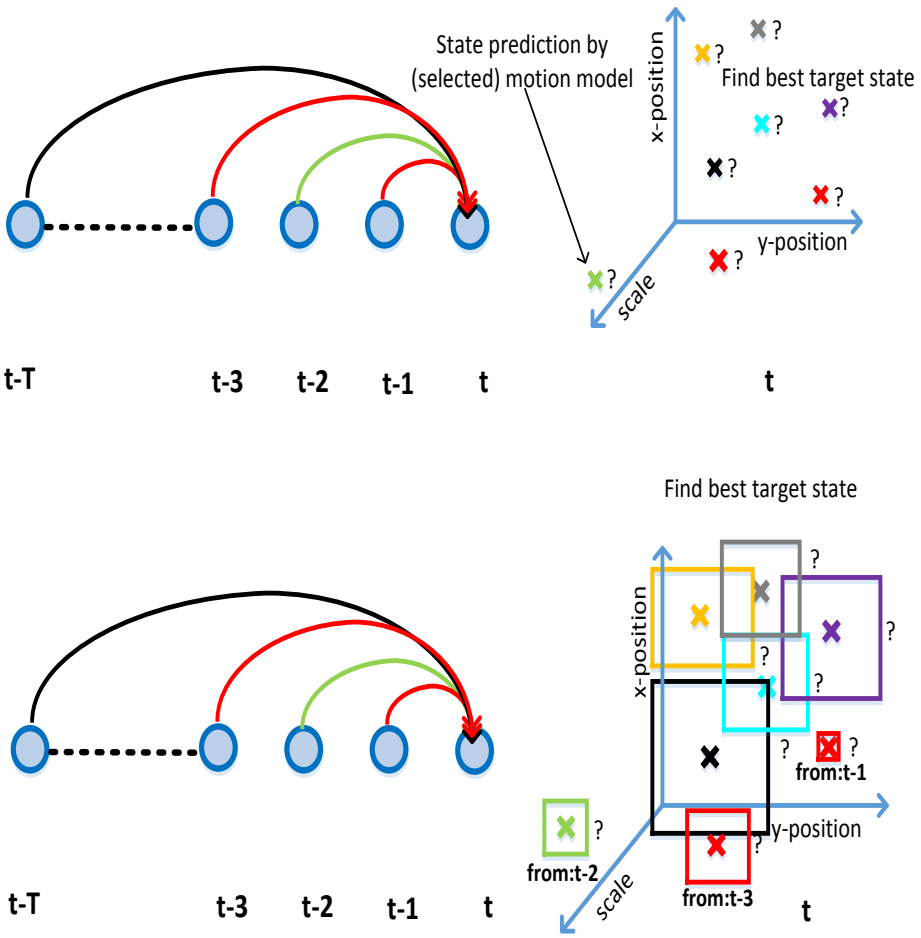


FIGURE 5.2: (Top row) There exists T (most suitable) state predictions at time t, and it is required to search for the best target state around these predictions. (Bottom row) The search is modelled by allocating each state prediction a certain area (cell).

The original instantiation of MTS extends bootstrap PF to allocate a fixed number of particles with a certain spread around each state prediction (as shown in Fig. 5.3) to obtain $\hat{\mathbf{X}}_t$. This spread is equal to the uncertainty associated with the corresponding state prediction computed as described above.

5.1.3 A Generalized WLMCMC sampling

WLMCMC sampling was introduced by [Kwon and Lee, 2008] to search for the target in a whole image after dividing it into a fixed grid of equal sized cells (see Fig. 5.4). It is composed of the Wang-Landau estimation and Markov Chain Monte Carlo (MCMC) method. Wang-Landau estimation is a Monte Carlo algorithm that was introduced in

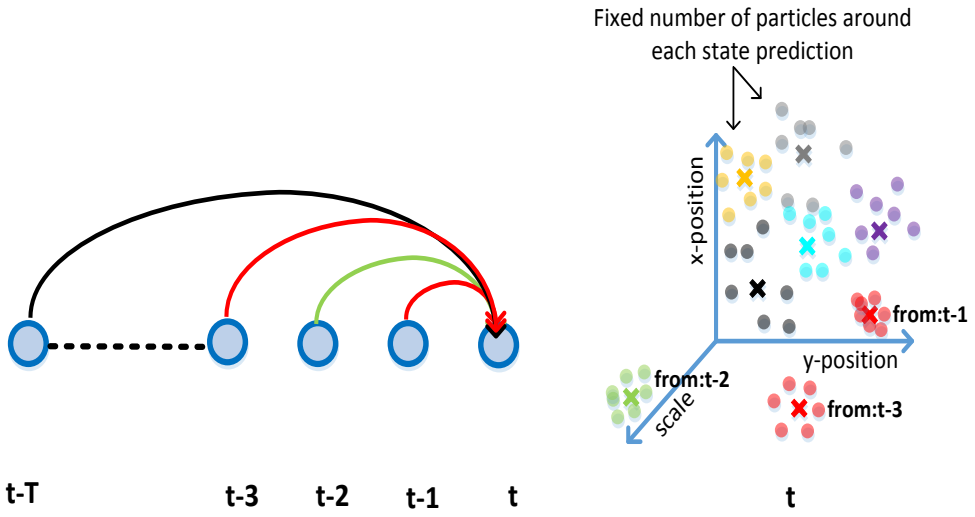


FIGURE 5.3: **Graphical representation of the search pattern in the original instantiation of MTS.** The original instantiation of MTS allocates fixed number of particles around each state prediction to estimate the best target state.

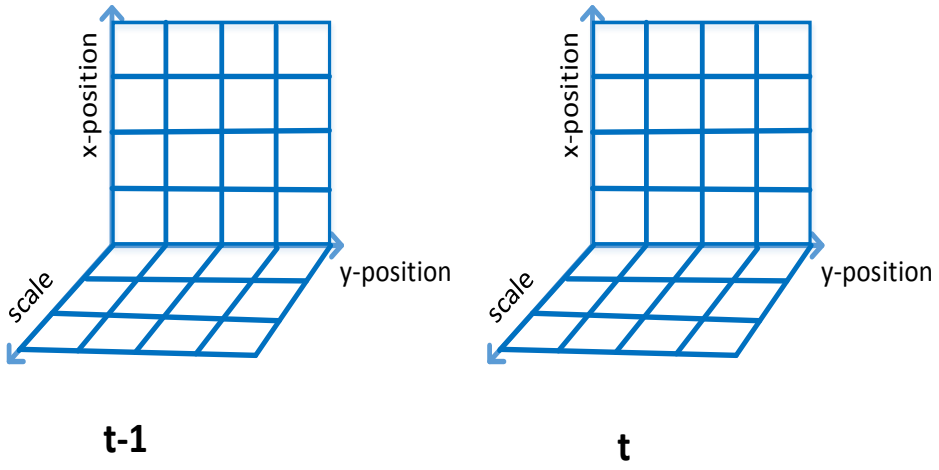


FIGURE 5.4: **Division of a state space into a fixed grid of equal sized cells.** A 3D state space divided into a fixed grid of equal sized cells at time t-1 and time t.

the physics literature for calculating the density of states (DOS) by performing a set of random-walks in different energy cells [Wang and Landau, 2001].

5.1.3.1 Wang Landau Monte Carlo (WLMC) method

The aim is to estimate the DOS score for every cell, where the DOS score is high for a cell if it contains highly probable target states. As it is intractable to accurately calculate the DOS score for every cell, the WLMC sampling approximately estimates it through Monte Carlo simulation [Wang and Landau, 2001]. The energy landscape of a large and/or high dimensional space may contain multiple competing local maxima [Zhou et al., 2012]. In such cases, it is extremely difficult to accurately calculate DoS due to the exponential growth of configuration states and a non-existence of analytic solution to the complex distribution over this large and/or high dimensional space. Sampling-based solutions such as Metropolis algorithm [Metropolis et al., 1953] are prone to trapping in local modes, and thus, results in poor sampling efficiency and convergence far from the true solution [Zhou and Bhatt, 2005],[Zhou et al., 2012]. WLMC algorithm provides an efficient way to approximately estimate it as it solves the DoS estimation and sampling problem in one go [Wang and Landau, 2001].

This method maintains a histogram h, and each bin of this histogram corresponds to a specific cell C_t^k. When C_t^k is visited, its bin count $h(C_t^k)$ is increased by 1, and its DOS score $g(C_t^k)$ is modified by multiplying by a modification factor $f > 1$.

$$g(C_t^k) \leftarrow g(C_t^k) * f, \tag{5.5}$$

where $g(C_t^k)$ is initially set to 1 for all k. As the simulation progresses, the random-walk generates a semi-flat histogram. A histogram is considered semi-flat if the value of the lowest bin is larger than 80% of the average value of all bins in h [Wang and Landau, 2001]. The semi-flat histogram denotes that the method has explored all the cells to at least some degree. Now the method performs the next random-walk in a coarse-to-fine manner to obtain more accurate DOS estimates. For this, the f factor is reduced to $f \leftarrow \sqrt{f}$ and the histogram is reset to $h = \{0, 0, 0, , |\mathbf{C}_t|\}$. Where $|\mathbf{C}_t|$ is the number of cells at time t, and is equal to the number of bins in histogram h. The method continues until the histogram becomes semi-flat again; then restarts the random-walk with a finer modification factor. The algorithm terminates when the modification factor becomes close to 1 or the number of iterations reaches a pre-defined value.

5.1.3.2 Proposal Step

The proposal step defines how the transition from the current state to a new state will occur based on some previous knowledge of target motion. In this case, the previous knowledge of target motion is that it can move from the current cell to any of the cells within one proposal step. The proposal density is defined as

$$Q(\mathbf{X}_t^{'}; \mathbf{X}_t) = Q_c(\mathbf{X}_t^{'}). \tag{5.6}$$

Q_c proposes a new state $\mathbf{X}_t^{'}$ in two stages. In the first stage, a cell C_t^k is chosen randomly from the T available cells to obtain diverse states. In the second stage, the x-location and y-location of $\mathbf{X}_t^{'}$ are uniformly drawn from the chosen cell C_t^k, and the scale part of $\mathbf{X}_t^{'}$ is proposed by adding zero-mean Gaussian noise with standard deviation $\sigma_s \times k$ to the scale part of $\hat{l}_t^k$. A Gaussian noise is added to the scale part of the predicted state $\hat{l}_t^k$ to propose a new scale for $\mathbf{X}_t^{'}$ since it is assumed that the scale does not vary much around the predicted state.

5.1.3.3 Acceptance Step

The acceptance ratio decides whether the proposed state is accepted or not using the likelihood ratio between the proposed state $\mathbf{X}_t^{'}$ and the current state $\mathbf{X}_t$

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{'})Q(\mathbf{X}_t; \mathbf{X}_t^{'})}{p(\mathbf{Y}_t|\mathbf{X}_t)Q(\mathbf{X}_t^{'}; \mathbf{X}_t)}\right], \tag{5.7}$$

The WLMCMC algorithm integrates the density of states term with the acceptance ratio in Eq. 5.7. Let D be a mapping function from the state $\mathbf{X}_t$ to the cell C_t^k, which contains the state $\mathbf{X}_t$.

$$D : X_t \rightarrow C_t^k \tag{5.8}$$

Then the acceptance ratio becomes

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{'})\frac{1}{g(D(\mathbf{X}_t^{'}))}Q(\mathbf{X}_t; \mathbf{X}_t^{'})}{p(\mathbf{Y}_t|\mathbf{X}_t)\frac{1}{g(D(\mathbf{X}_t))}Q(\mathbf{X}_t^{'}; \mathbf{X}_t)}\right], \tag{5.9}$$

where $g(D(\mathbf{X}_t^{'}))$ denotes the density of states in the cell which contains $\mathbf{X}_t^{'}$.

Eq. 5.9 has two important advantages over Eq. 5.7. The first is that it provides a systematic way for a Markov Chain to escape local maxima and capture the global

maximum. This is required because the cells could be positioned far apart from each other in state space. And in these situations, the Markov Chain has a higher probability of meeting local maxima. The second advantage of Eq. 5.9 is that it enables the Markov Chain to spend more time around local maxima, while guaranteeing to explore all the cells to some degree. Again, this is desirable in this scenario because there could be any number of cells containing highly probable target states, and this should be discovered by visiting each of them to some degree.

The DOS score $g(D(\mathbf{X}_t^{'}))$ in Eq. 5.9 is calculated exactly in the same way as described in subsection 5.1.3.1. For instance, if a state proposed by Eq. 5.6 is accepted by the acceptance ratio in Eq. 5.9, and the state belongs to cell C_t^k, then the DOS score of the cell $g(C_t^k)$ is modified by the factor f, and its bin count $h(C_t^k)$ is increased by 1. Otherwise, the same procedure is applied to the cell which contains the current state. Fig. 5.5 graphically illustrates the steps involved in a single iteration of generalized WLMCMC, and Algorithm 2 details relevant steps of the generalized WLMCMC method given variable sized cells, where each cell is formed around a certain state prediction.

Algorithm 2 Generalized WLMCMC method for variable sized cells.

Input: A set of variable sized cells at time t: $\mathbf{C}_t = \{C_t^k | k = 1, ...T\}$
Output: Best state of the target at time t: $\hat{\mathbf{X}}_t$

Initialize the DOS score for each cell $g(C_t^k) = 1$, and the bin count for each cell $h(C_t^k) = 0$, where $k = 1, ..., T$.
Set $f = 2.7$
for $q = 1$ *to* N, where N is the total number of particles
- Given the current state $\mathbf{X}_t^q$, propose a new state $\mathbf{X}_t^{'}$ using the Eq. 5.6.
- Use Eq. 5.9 to compute the acceptance ratio.
- if the proposed state is accepted then set $\mathbf{X}_t^{q+1}$ to $\mathbf{X}_t^{'}$, else set $\mathbf{X}_t^{q+1}$ to $\mathbf{X}_t$.
- $g(D(\mathbf{X}_t^{q+1})) \leftarrow g(D(\mathbf{X}_t^{q+1})) * f$
- $h(D(\mathbf{X}_t^{q+1})) \leftarrow h(D(\mathbf{X}_t^{q+1})) + 1$
- If h is semi-flat then reset $h(C_t^k) = 0, \forall k$ and $f \leftarrow \sqrt{f}$.

end
Compute the best state $\hat{\mathbf{X}}_t$ using Eq. 5.2.

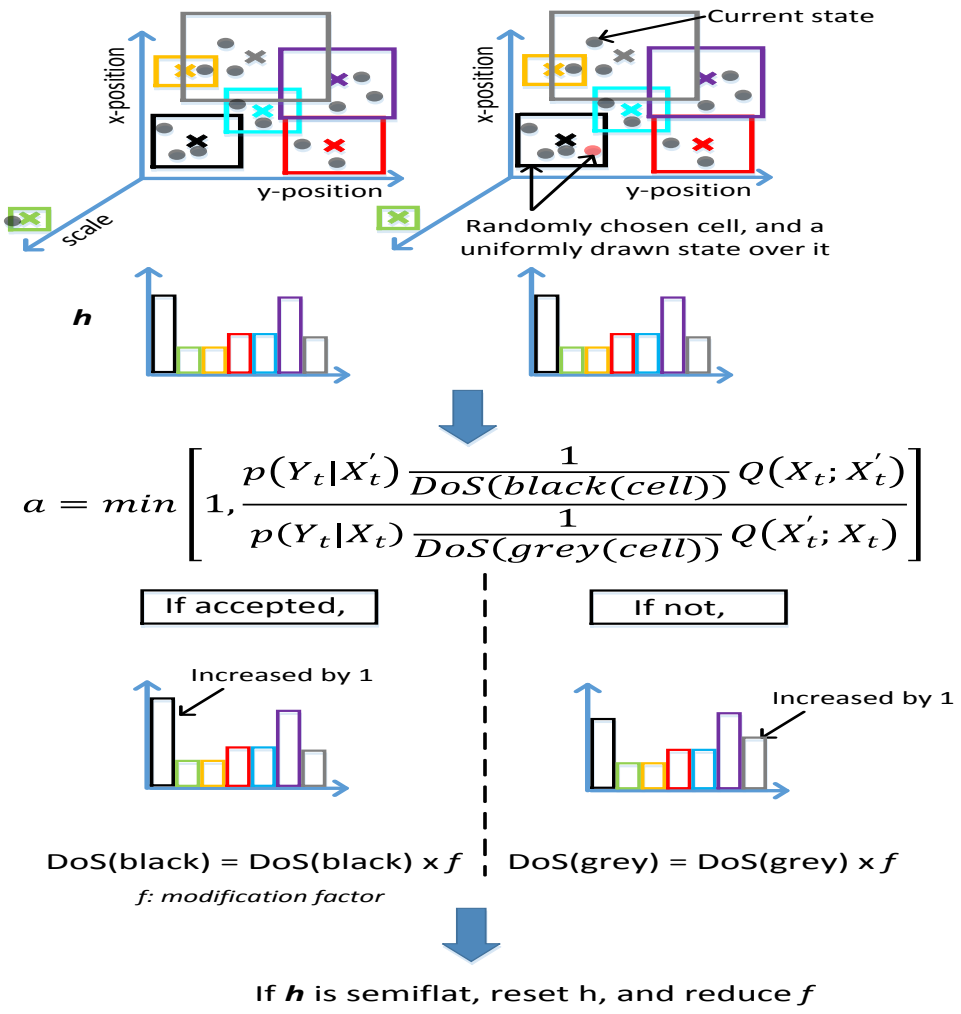


FIGURE 5.5: **Graphical illustration of the steps involved in a single iteration of generalized WLMCMC.**

5.1.4 Experimental Details and Results

5.1.4.1 Data

Eleven video sequences were used for the experimental evaluation. Seven are publicly available (*PETS 2001 Dataset 1*[1], *TUD-Campus*[Andriluka et al., 2008], *TUD-Crossing*[Andriluka et al., 2008], *Person*[Dihl et al., 2011], *car*[Wu et al., 2013], *Person2*[2], and *PETS 2009 Dataset S2*[3]) and four are our own (*squash*, *ball1*,*ball2*, and *toy1*).

[1] *PETS 2001 Dataset 1* is available from http://ftp.pets.rdg.ac.uk/
[2] *Person2* is available from http://www.iai.uni-bonn.de/ kleind/tracking/
[3] *PETS 2009 Dataset S2* is available from http://www.cvg.rdg.ac.uk/PETS2009/

5.1.4.2 Experimental Settings

The multiple temporal scale framework (MTS) based on generalized WLMCMC is denoted by MTS-GWL. MTS-GWL was compared to the original instantiation of MTS, which is based on the bootstrap Particle Filter and is denoted by MTS-PF, and seven state-of-the-art trackers. The state-of-the-art trackers are WLMCMC [Kwon and Lee, 2008], Visual Tracking Decomposition (VTD) [Kwon and Lee, 2010], Fragment-based Tracker (FragT) [Adam et al., 2006], Sparsity-based Collaborative Model (SCM)[Zhong et al., 2012], Adaptive Structural Local Sparse Appearance Model (ASLA)[Jia et al., 2012] Real-Time Robust L1-Tracker using Accelerated Proximal Gradient (L1-APG) [Bao et al., 2012], and Semisupervised Boosting Tracker (Semi) [Grabner et al., 2008].

In terms of search methods, MTS-PF, SCM, ASLA and L1-APG are based on particle filters, FragT and Semi utilize dense sampling methods, WLMCMC and VTD use MCMC. The minimum and maximum number of samples used for WLMCMC, VTD, SCM, ASLA, and L1-APG was 600 and 640, respectively. The minimum and the maximum size of the cell in terms of half width and half height in image space were 1 pixel and 30 pixels, respectively. For MTS-PF and MTS-GWL, model-scales of 2,3,4, and 5 frames were used, and at each model-scale a linear motion model was learned. MTS-PF and MTS-GWL utilized the HSV colour histogram as the observation model and Bhattacharyya coefficient as the distance measure [Pérez et al., 2002]. The list of tracking parameters is given in Appendix A.

5.1.4.3 Experimental Results

Table 5.1 reports tracking accuracy of 9 trackers on 11 video sequences in terms of centre location error (in pixels), percentage of correctly tracked frames based on Pascal score [Santner et al., 2010], and precision at a fixed threshold of 20 pixels. The first nine videos involve occlusions of varying lengths, and the last two contain both occlusions and rapid motion variations.

According to center location error criterion, out of eleven sequences, MTS-GWL improved the accuracy of MTS-PF in four sequences and it revealed the same performance as MTS-PF in four sequences. Likewise, in terms of Pascal score and precision, MTS-GWL performed better than MTS-PF in eight out of eleven sequences. It is important to note that both the methods use same number of particles, and the only difference between them is the search method. Given multiple cells of variable size formed around state predictions, MTS-GWL produces more samples from the cells with a higher probability of containing local maxima that increases the likelihood of reaching the global

TABLE 5.1: **Tracking accuracy of 9 trackers on 11 video sequences**

(A) Mean centre location error in pixels is given, averaged over all frames of all videos showing occlusions. Each tracker was run five times and the results were averaged. The best results are marked in bold. N_t is the number of particles used in MTS-GWL and MTS-PF.

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	MTS-PF	MTS-GWL	N_t
ball2	76	71	71	66	78	106	37	**17**	19	640
TUD-Camp	186	180	100	186	61	112	**22**	24	**21**	180
TUD-Cross	**2**	6	**2**	63	62	5	50	25	25	500
PETS 2001	61	63	60	83	114	67	90	25	**17**	680
Person	91	80	103	85	177	84	25	10	**10**	400
PETS 2009	35	13	81	94	29	10	91	**7**	8	280
car	**8**	31	31	47	38	15	28	25	25	400
toy1	88	85	111	98	99	107	30	**21**	21	600
jogging	110	104	45	70	30	94	**19**	25	25	400
squash	40	34	60	20	68	35	22	12	**10**	100
ball1	91	96	124	69	66	188	23	15	**14**	280

(B) A(B): A - the percentage of correctly tracked frames based on Pascal Score [Santner et al., 2010]; B - Precision at a fixed threshold of 20 pixels. Pascal score is computed by assessing to what extent the tracking template overlaps the ground truth template as a ratio. If the Pascal score is greater than 0.5 in a certain frame, that frame is counted as a correctly tracked frame. Precision is computed by dividing the number of frames, where estimated target location was not beyond the fixed threshold distance of 20 pixels of the ground truth, by the total number of frames in a video sequence. The best results are marked in bold.

Sequence	SCM	ASLA	L1	VTD	Semi	FragT	WL	MTS-PF	MTS-GWL
ball2	12(0.21)	9(0.17)	7(0.21)	9(0.11)	7(0.13)	9(0.09)	28(0.53)	31(0.8)	**36**(**0.82**)
TUD-Camp	14(0.17)	10(0.14)	19(0.21)	25(0.25)	38(0.34)	27(0.27)	46(**0.61**)	55(0.57)	**64**(0.6)
TUD-Cross	**100**(**1**)	99(0.9)	**100**(**1**)	24(0.23)	41(0.42)	87(1)	25(0.23)	61(0.59)	63(0.6)
PETS 2001	23(0.33)	23(0.27)	22(0.25)	20(0.25)	17(0.2)	16(0.31)	19(0.52)	58(0.65)	**66**(**0.78**)
Person	45(0.46)	44(0.45)	10(0.12)	43(0.45)	20(0.2)	38(0.41)	49(0.86)	79(0.93)	**79**(**0.93**)
PETS 2009	26(0.36)	36(0.7)	21(0.26)	21(0.21)	27(0.45)	65(0.73)	7(0.23)	**70**(**0.97**)	53(0.96)
car	**92**(**0.93**)	62(0.64)	66(0.65)	66(0.65)	55(0.46)	80(0.76)	62(0.52)	71(0.72)	71(0.72)
toy1	18(0.19)	19(0.2)	15(0.15)	16(0.18)	16(0.18)	3(0.09)	**49**(0.8)	43(0.8)	46(**0.82**)
jogging	21(0.22)	22(0.22)	21(0.21)	22(0.22)	**60**(**0.61**)	21(0.21)	42(**0.61**)	20(0.44)	20(0.41)
squash	60(0.62)	38(0.56)	9(0.11)	68(0.78)	44(0.7)	37(0.5)	50(0.75)	71(0.92)	**79**(**0.97**)
ball1	6(0.06)	3(0.04)	2(0.05)	19(0.22)	19(0.33)	2(0.02)	35(0.79)	40(0.83)	**41**(**0.89**)

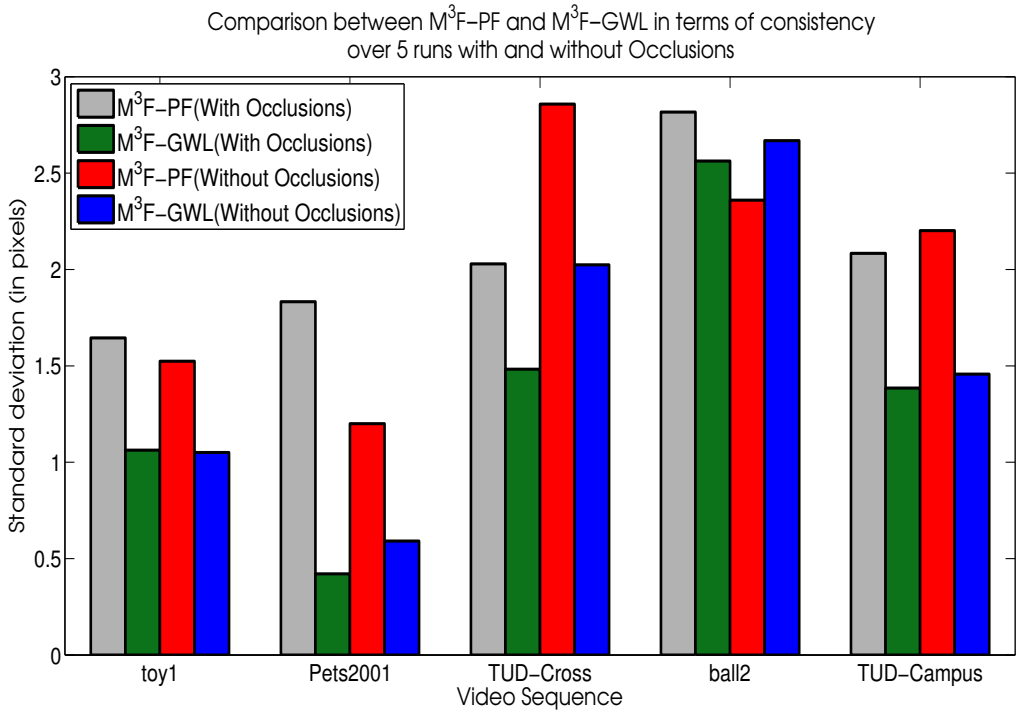


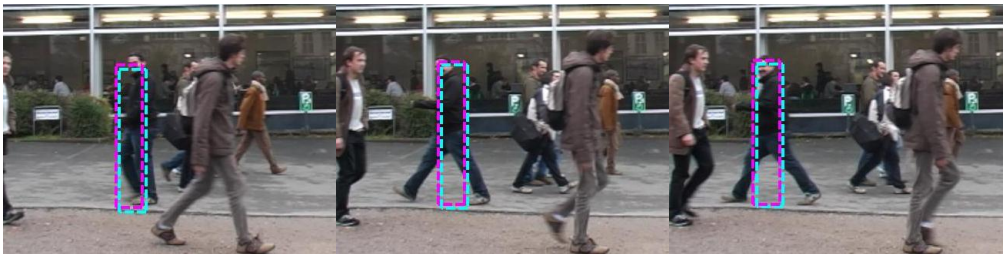
FIGURE 5.6: **Comparison of tracking consistency over five runs between MTS-GWL and MTS-PF with and without occlusions.** Each bar in this figure is a standard deviation (in pixels) calculated over a set of five mean center location errors (in pixels). Each mean value in this set is computed by averaging the centre location error over all frames (with or without occlusion) for a video sequence.

maximum. In other words, the method provides a statistical way to reach global optimum of the likelihood function through combining efficient multiple-range random-walks with the MCMC method in a principled way. On the other hand, MTS-PF just allocates a fixed number of particles with a certain spread around each state prediction, and thus so is more likely to miss the global maximum.

Fig. 5.6 shows a comparison between MTS-PF and MTS-GWL in terms of tracking consistency over five runs with and without occlusions on five different sequences. The improved consistency of MTS-GWL over MTS-PF under both situations suggests that while the proposed framework has the potential to handle occlusions, its tracking consistency can be improved further with a sophisticated search method such as generalized WLMCMC.

The target faces occlusions of different lengths in the *ball2* sequence. Fig. 5.7(a) shows tracking results of MTS-PF, and MTS-GWL in this sequence. Although both the methods recover the target after occlusion (frame # 128, and frame # 129), MTS-GWL provides more accurate tracking than MTS-PF. Similarly, the target is occluded twice in the *TUD-Campus* sequence. MTS-GWL is slightly more accurate than MTS-PF in tracking the target as shown in Fig. 5.7(b).

(A) frame # 128, 129, and 216 of *ball2* sequence.



(B) frame # 21, 26, and 28 of *TUD-Campus* sequence.

(C) frame # 32, 92, and 282 of *squash* sequence.

FIGURE 5.7: Tracking results of MTS-PF(cyan) and MTS-GWL(magenta) in *ball2*, *TUD-Campus*, and *squash* sequences.

It is difficult to maintain accurate tracking when a target undergoes both abrupt motion variations and shorter occlusions. Fig. 5.7(c) displays tracking results of MTS-PF, and MTS-GWL in the *squash* sequence. MTS-GWL achieves better accuracy than MTS-PF in tracking a target with large motion uncertainty (frame # 32, frame # 92, and frame # 282).

5.1.5 Discussion

Thus far, a search method has been generalized to find the best possible target state around multiple competing hypotheses. These hypotheses are state predictions generated in the proposed multiple temporal scale framework (MTS). The search is modelled

by assigning a certain area in state space, which is called a cell, to each state prediction. To search for the best target state in these cells, WLMCMC sampling is generalized to cells of variable size and location.

Experimental evaluation showed that the generalized search method improves the performance of the proposed framework to some extent compared to its original instantiation, which is based on the extension of Particle Filter. The generalized search method enhances performance since it is more principled than its counterpart in finding the global optimum of the likelihood function.

On the other hand, it is also important to understand why there was not a substantial improvement in performance. The success of the search component in MTS is largely dependent on the locations of state predictions in the state space. If a majority of the predictions are completely off-target i.e. far away from the true modes of the target distribution, and out of the remaining ones, none of them is located very close to the true target state, then the search method might not be able to reach the global solution. In cluttered environments these off-target predictions can be even more threatening, if they are located close to an object whose appearance is very similar to the tracked target.

The success of a search method is also contingent on the appearance model. Since MTS uses a fixed, and a very simple appearance model, it might not stay valid when the target appearance is varying. In such cases, despite the accurate state predictions, the search method would never be able to estimate the true underlying posterior distribution, and would result in a sub-optimal solution.

For the search method to converge to the true underlying posterior distribution, the state predictions should lie close to the true modes of this distribution, and the appearance model should match closely the true target appearance. Therefore, the role of MTS, model selection criterion is increasingly important. Recall, the current selection criterion in MTS picks one model (prediction) from each of the several previous time-steps using a visual likelihood score. It is simple and efficient, but ignores the fact that predictions from some previous time-point might be inaccurate and should be rejected completely.

To be able to devise a new model selection criterion, it is important to first underline the dimensions of (motion) model space and choose some subset of variables to simplify the model selection problem. In this regard, Section 5.2 explores MTS further by first developing an understanding of the model selection problem, and then proposing a new selection criterion.

5.2 A New Model Selection Criterion

Tracking frameworks employing multiple motion models such as MTS may generate multiple competing hypotheses (state predictions) at a given time. These state predictions can also be considered motion priors that guide search toward the correct modes of the target distribution. However, these state predictions are not all equally accurate; some of them might lie in non-target regions. Under these circumstances, it might be appropriate to reject completely off-target predictions, find structure in those remaining, and utilize representative ones to define search regions. Ideally, these search regions should be close to the local modes of the target distribution, far away from the distractors, and small in number.

This section investigates the model selection problem in the context of MTS, and develops a new motion model selection criterion. This is achieved by exploring the possible dimensions of the motion model space to develop an understanding of the model selection problem. Then, a new selection criterion is developed.

5.2.1 Model Selection Problem

In modern scientific enterprise, model selection is the task of choosing a statistical model from a set of candidate models, given data. It is an important task because it saves computer and analyst time. A general principle for selection is given by Occam's razor [Blumer et al., 1987]. This states that given candidate models of similar explanatory power, the model with minimum complexity is most likely the best to select.

A model selection criterion is expected to balance goodness-of-fit (how well the model fits the data) with simplicity (small number of degrees of freedom). More complex models will better fit the data than any other in the candidate set, but the additional parameters may not reveal anything of interest. For instance, a 5th order polynomial will exactly fit 4 points, but those 4 points might be randomly distributed around a straight line. Keeping in view the importance of model selection, many selection criteria have been proposed. Among several methods proposed for model selection, Akaike Information Criterion (AIC) [Akaike, 1974], Bayesian Information Criterion (BIC) [Schwarz et al., 1978], Bayesian Model Averaging (BMA) [Madigan and Raftery, 1994], and Bayes Factor (BF) [Kass and Raftery, 1995] are the most well-known.

The model selection problem to be discussed here has similar purpose i.e. to deselect poor motion models, whose predictions are located far away from the true target state, but the nature of the problem is different. In the proposed framework (MTS), T sets of motion models are available at time t, one from each of the T preceding time-steps.

Each set of models at time t is represented by its corresponding set of predictions. Every prediction is a point in a 3D state space composed of position and scale. Each available model (prediction) at time t has three associated variables: x, y position and scale s.

In fact, it is important to note that each motion model was generated over an appropriate model-scale at some previous time-step. This means that there are a further three variables associated with each motion model available at the current time-step t. These three additional variables are: model-scale, prediction-scale, and prediction-history. Model-scale is the sequence of states over which a model was generated, and prediction-scale is the previous time point at which this model was learned e.g. $[t-3]$. Prediction history is a sequence of future predictions made by a model before approaching the current time t (e.g. a model learned at time $[t-3]$ would have predicted target state at time $[t-2]$, and $[t-1]$ before time t). Fig. 5.8 illustrates the three additional variables. With this, the model space in the model selection problem has six dimensions at time t: x, y position, scale s, model-scale, prediction-scale, and the prediction-history. Now, the question is how to select a subset of models from this model space that are located close to the true target state, and what should be the cardinality of this subset.

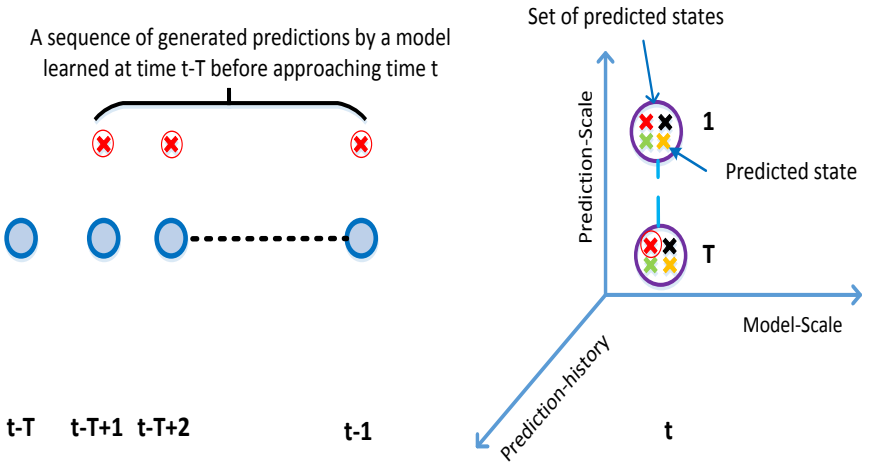


FIGURE 5.8: Graphical Illustration of three additional dimensions of model space in the model selection problem.

In the initial attempt, considering the full model space (all six dimensions) for the development of a new model selection criterion might be a complicated task. The next section picks three dimensions, position and scale, to reduce the complexity of the problem.

5.2.2 A New Motion Model Selection Criterion

This section proposes a new motion model selection strategy, utilizing the locations of predictions made by available motion models at the current time t in 3D state (position and scale) space. The aim is to define search regions that are close to the true target state.

Selection is achieved in two steps. In the first step, state predictions with very low visual likelihood score are rejected quickly by an automatically chosen threshold. This threshold is generated by examining the density of the variable that represents visual likelihood score of the state predictions corresponding to available motion models. In the second step, the remaining state predictions are spatially clustered to identify possible prediction classes. To model search, two areas in state space, called cells, are assigned to each cluster. One is associated with the cluster center and, the other to the member of this cluster with highest visual likelihood score. The size of these cells is proportional to the within cluster standard deviation. Creation of two cells per cluster allows the proposed criterion to exploit both the motion and the appearance cues to capture local maxima of the target distribution.

Let $\mathbf{U}_t = \{\mathbf{u}_t^q | q = 1, ..., G \times T\}$ be the set of state predictions available at time t, where $G = |\mathbf{M}_t|$ is the number of motion models learnt at each (previous) time-step, and T is the number of prediction-scales. Each $\mathbf{u}_t^q$ is weighted using the observation model, and the corresponding likelihood scores are represented by $\mathbf{V}_t = \{\mathbf{v}_t^q | q = 1, ..., G \times T\}$.

5.2.2.1 Rejecting Low Visual Likelihood State Predictions

This step is based on the assumption that not all the state predictions are accurate enough, and should not be searched for the target state. It filters out state predictions that have a very low visual likelihood score with an automatically selected threshold. This threshold is decided by examining the density of a random variable, whose examples are visual likelihood scores of state predictions. Note that visual likelihood score is chosen over other cues like motion for this task because it is considered most reliable (in terms of producing false positives).

Let $p(\mathbf{v}_t) = \frac{1}{Zh} \sum_{q=1}^{Z} K(\frac{\mathbf{v}_t - \mathbf{v}_t^q}{h})$ be the kernel density estimate (KDE) of the (one-dimensional) random variable $\mathbf{V}_t$, whose examples are visual likelihood scores of state predictions. Where $K(.)$ is the Gaussian kernel, h is the bandwidth, and $Z = G \times T$. Fig. 5.9 shows an example KDE of a random variable that is a mixture of three normal distributions.

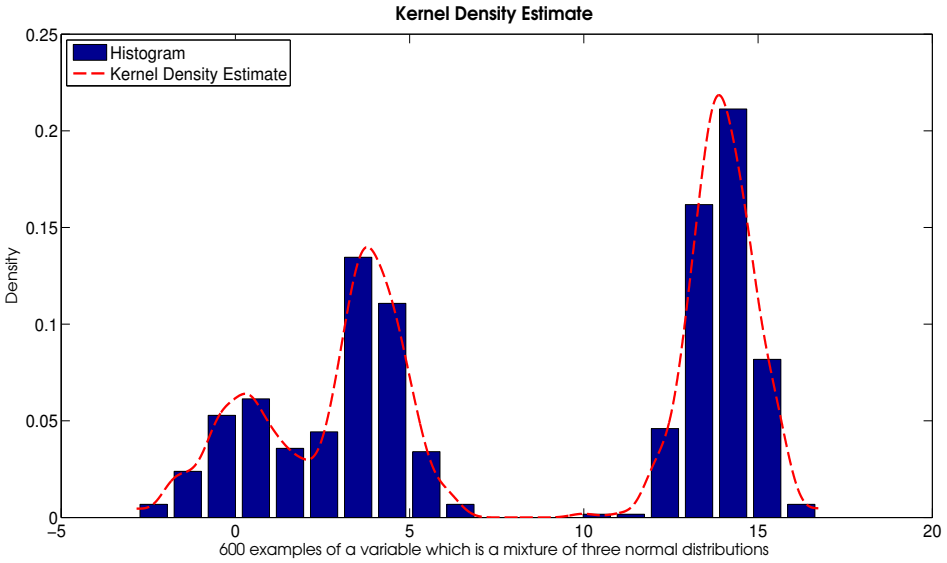


FIGURE 5.9: Kernel density estimate and histogram of a random variable which is a mixture of three normal distributions.

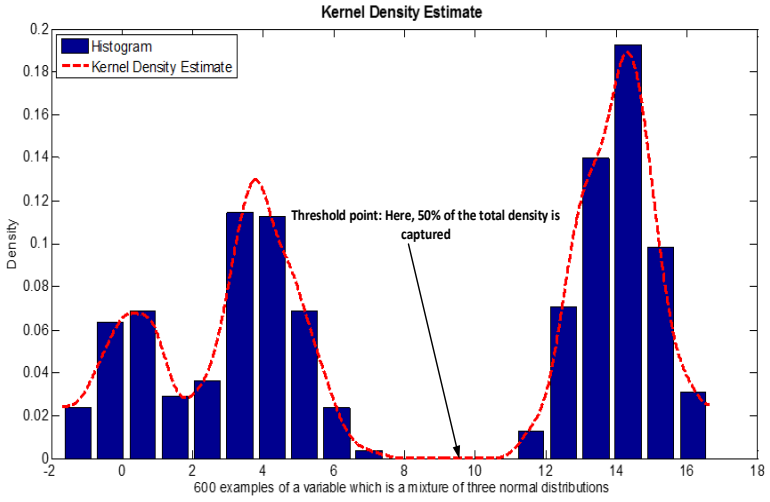


FIGURE 5.10: Threshold point selected on capturing 50% of the total density of a random variable, which is a mixture of three normal distributions.

A very simple procedure is adopted to select a threshold η_t for rejecting the state predictions. Given the KDE of $\mathbf{V}_t$, take the point along the x-axis as η_t at which e% of the density is captured after starting from the maximum value of $\mathbf{V}_t$. e is the fraction of the total density of random variable $\mathbf{v}_t$ to be considered for selecting the threshold η_t. It lets the selection of η_t by taking into account the shape of the density associated with the (random) variable $\mathbf{v}_t$. That is, as $p(\mathbf{v}_t)$ can take arbitrary shapes, the aim is to select a threshold point representative of this density. Fig. 5.10 illustrates the threshold

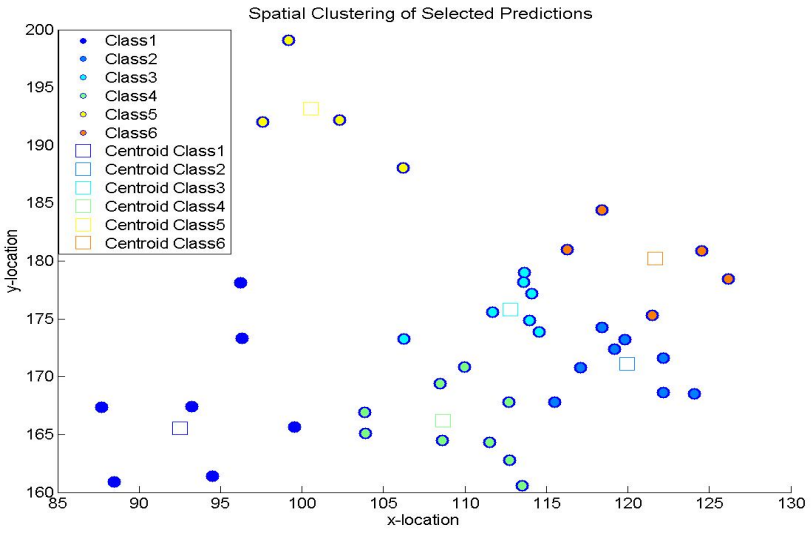


FIGURE 5.11: Six different prediction classes identified by K-Harmonic Means (KHM) in $\mathbf{D}_t$. Only x and y locations of the target state are shown here.

point selected to capture 50% of the total density of a random variable.

Given threshold η_t, $\mathbf{v}_t^q$, which is the q_{th} instance of $\mathbf{V}_t$, is rejected if

$$\mathbf{v}_t^q < \eta_t. \tag{5.10}$$

Otherwise it is selected. The set of selected predictions generated by this procedure is denoted by $\mathbf{D}_t$.

5.2.2.2 Spatial Clustering

The aim of spatial clustering is to discover possible prediction classes in the set of selected predictions $\mathbf{D}_t$. These potential prediction classes will then lead towards the establishment of search regions of different sizes in the state space.

The K-Harmonic Means (KHM) method is used to cluster $\mathbf{D}_t$ into K prediction classes, where $K = \sqrt{Z}$. KHM is chosen over K-means as it is insensitive to the initialization of centers [Zhang et al., 2001]. Recall, each instance of $\mathbf{D}_t$ is a state prediction $\mathbf{l}_t^k(x_t, y_t, s_t)$. Fig. 5.11 displays 6 different motion classes (clusters) discovered by KHM in $\mathbf{D}_t$.

Within-cluster-variance is a measure of compactness of a cluster. Here, it is interpreted as how confident the predictions belonging to this cluster are about the true target state according to the motion cue. In other words, if the predictions are tightly clustered, then there is a high likelihood of target being present according to motion information in the vicinity of these predictions.

5.2.2.3 Formation of Search Regions

The set of cluster centers identified in section 5.2.2.2 is denoted by $\mathbf{\Omega}_t = \{\Omega^1, ..., \Omega^K\}$, where K is the total number of clusters. The search is modeled by assigning each cluster center Ω^m, and the member (state prediction) of this cluster having maximum visual likelihood score a certain area in the state space, which is called a cell. The size of this cell is proportional to the sum of within-cluster-standard-deviation of this cluster and the standard deviation of the zero mean Gaussian noise acting on target state between two consecutive time-steps. For instance, the cell size around the center of m_{th} cluster and the (visually) highest weighted member of this cluster would be $2 \times (STD(\mathbf{D}_t, \Omega^m_x) + \sigma_x)$, and $2 \times (STD(\mathbf{D}_t, \Omega^m_y) + \sigma_y)$. Along the third dimension, scale, the uncertainty would be $(STD(\mathbf{D}_t, \Omega^m_s) \times \gamma^4) + \sigma_s$. Where $STD(\mathbf{D}_t, \Omega^m_x)$, and $STD(\mathbf{D}_t, \Omega^m_y)$, and $STD(\mathbf{D}_t, \Omega^m_s)$ return the within-cluster-standard-deviation along x, y, and s dimensions of data, $\mathbf{D}_t$, that belongs to cluster centered on Ω^m. σ_x, σ_y, and σ_s are the standard deviations of the zero mean Gaussian noise acting on x, y, and s part of the target state between two consecutive time-steps, respectively. Algorithm 3 details relevant steps of this new motion model selection criterion.

Note that the formation of cells around both the cluster center and the (visually) highest weighted member of this cluster allows the proposed criterion to exploit both motion and appearance information to locate local modes of the target distribution. The cell size is linked to the within-cluster-deviation of a cluster since this measure reflects motion uncertainty around the center of the cluster.

With K clusters, the total number of cells are $2 \times K$. The set of these variable sized cells is represented by $\mathbf{C}_t = \left\{ C^f_t \,|\, f = 1, ...2 \times K \right\}$. To search for the best target state $\hat{\mathbf{X}}_t$ in these cells, the generalized Wang-Landau Markov Chain Monte Carlo proposed in section 5.1 is used. Search for the optimal target state is also possible by initializing a local optimizer such as the Mean Shift algorithm [Comaniciu et al., 2000], or initiating a Markov Chain such as the Metropolis Hastings algorithm [Hastings, 1970] at the center of each cell. However, these local search methods can be computationally expensive. In addition, their application in this scenario would not allow the comparison of this new selection criterion with the one used in section 5.1.

[4] γ is a weighting factor, and is empirically set to 0.01 in all the experiments.

Algorithm 3 A New Motion Model Selection Criterion

Input: A set of state predictions at time t: $\mathbf{U}_t = \{\mathbf{u}_t^q | q = 1, ..., G \times T\}$, and their visual likelihood scores at time t: $\mathbf{V}_t = \{\mathbf{v}_t^q | q = 1, ..., G \times T\}$.

Output: A set of variable sized cells at time t: $\mathbf{C}_t = \left\{C_t^f | f = 1, ...2 \times K\right\}$, where K is the number of clusters.

Rejecting low visual likelihood state predictions:
- Estimate the probability density function (PDF) of $\mathbf{V}_t$ using KDE.
- Choose threshold η_t as the point along the x-axis of PDF at which e% of the density is captured after starting from the maximum value of $\mathbf{V}_t$.
- Threshold $\mathbf{V}_t$ to get the set of selected predictions $\mathbf{D}_t$.

Spatial Clustering:
- Cluster $\mathbf{D}_t$ into K motion classes (clusters). The set of cluster centers is represented by $\mathbf{\Omega}_t = \{\Omega^1, ..., \Omega^K\}$.

Formation of Search Regions:

for $m = 1$ *to* K, where K is the total number of cluster
- Allocate a cell to Ω^m (the cluster center).
- Allocate a cell to the (visually) highest weighted prediction of the cluster centered at Ω^m.

end
- Return the set of variable sized cells $\mathbf{C}_t = \left\{C_t^f | f = 1, ...2 \times K\right\}$.

5.2.3 Experiments and Results

5.2.3.1 Image Sequences used for Evaluation

Eleven video sequences were used for experimental evaluation. Out of eleven, seven are publicly available (*PETS 2001 Dataset 1*[5], *TUD-Campus*[Andriluka et al., 2008], *TUD-Crossing*[Andriluka et al., 2008], *Person*[Dihl et al., 2011], *car*[Wu et al., 2013], *Person2*[6], and *PETS 2009 Dataset S2*[7]) and remaining four are our own (*squash*, *ball1*,*ball2*, and *toy1*).

5.2.3.2 Description of Trackers and Experimental Settings

The multiple temporal scale framework (MTS) based on the new model selection criterion is denoted by MTS-MS. MTS-MS is compared to the MTS-GWL, which is an instantiation of the MTS based on the existing model-set reduction method. Note that the only difference between MTS-MS, and MTS-GWL is the way the model selection takes place, and they both utilize generalized WLMCMC for searching the best target state.

[5] *PETS 2001 Dataset 1* is available from http://ftp.pets.rdg.ac.uk/
[6] *Person2* is available from http://www.iai.uni-bonn.de/ kleind/tracking/
[7] *PETS 2009 Dataset S2* is available from http://www.cvg.rdg.ac.uk/PETS2009/

For MTS-MS and MTS-GWL, the minimum and the maximum size of the cell in terms of half width and half height in image space were 1 pixel and 30 pixels, respectively, and model-scales of 2,3,4, and 5 frames were used, and at each model-scale a linear motion model was learned. MTS-MS and MTS-GWL utilized the HSV colour histogram as the observation model and Bhattacharyya coefficient as the distance measure Pérez et al. [2002]. The value of e, which is the percentage of total density to be captured for selecting threshold η_t, was set to 50.

5.2.3.3 Results

Fig. 5.12 reports performance comparison between MTS-MS and MTS-GWL in terms of percentage of correctly tracked frames based on Pascal score. Overall, the difference in the accuracy of two methods is not substantial in almost all sequences. However, MTS-MS performed better than MTS-GWL in four out of eleven sequences, whereas MTS-GWL had higher accuracy than MTS-MS in five out of eleven sequences. Both MTS-MS and MTS-GWL performed equally well in two sequences.

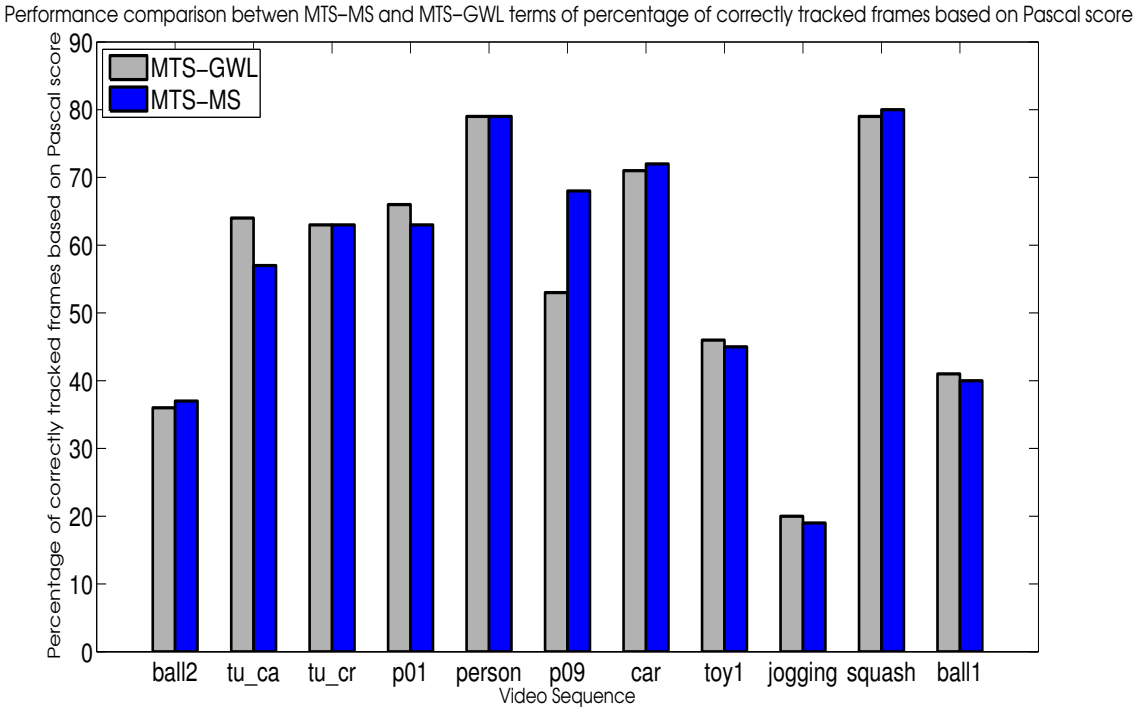


FIGURE 5.12: Performance comparison between MTS-MS and MTS-GWL in terms of percentage of correctly tracked frames based on Pascal score.

Fig. 5.13 plots performance comparison between MTS-MS and MTS-GWL in terms of precision at a fixed threshold of 20 pixels. Again, there is not a considerable difference in the accuracy of two methods in almost all sequences, but MTS-GWL showed slightly

improved performance than MTS-MS in seven out of eleven sequences. MTS-MS had only little higher accuracy than MTS-GWL in two out of eleven sequences.

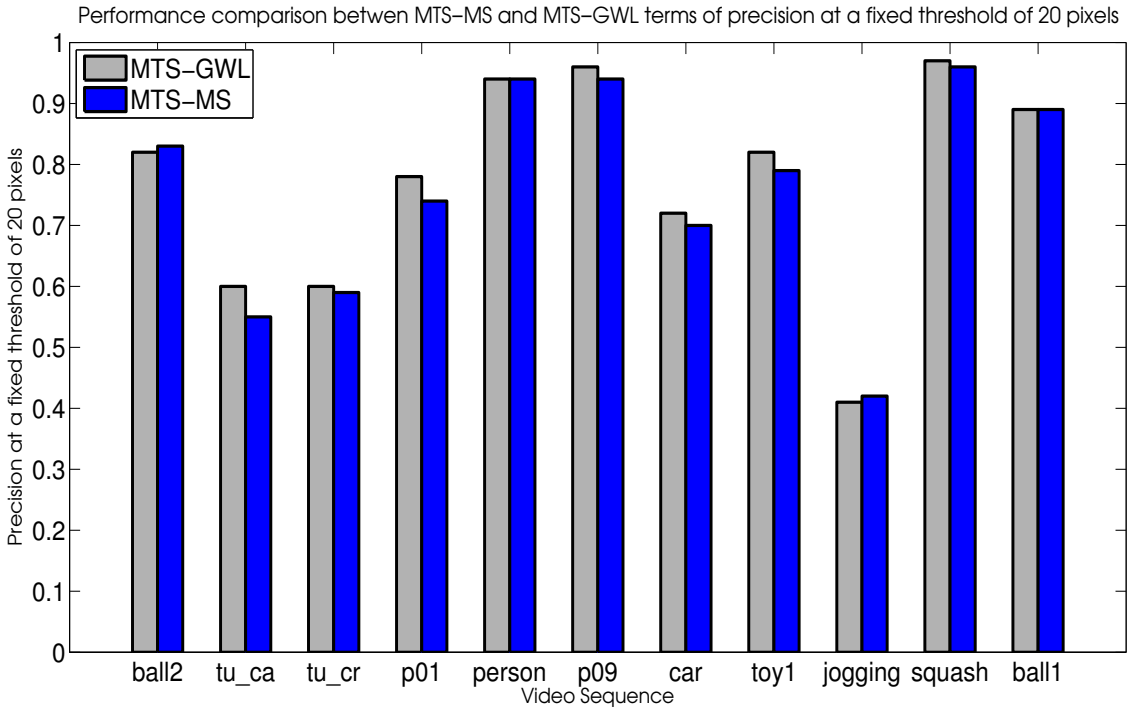


FIGURE 5.13: Performance comparison between MTS-MS and MTS-GWL in terms of precision at a fixed threshold of 20 pixels.

Although there was not a substantial difference in the performance of MTS-GWL and MTS-MS, MTS-GWL performed slightly better than MTS-MS in most of the sequences. This might be due to the fact that while rejecting models, the new selection criterion employed in MTS-MS takes into account just the appearance information, which is based on a very simple appearance model. Besides, it ignores the prediction-scale information attached to every model, which might somehow compensate for the errors in the appearance information. In contrast, the existing selection criterion used in MTS-GWL, naively capitalizes prediction-scale information. It selects one model from each of the previous time-points, and is not fully reliant on the appearance information.

MTS-MS was analyzed by varying the values of e%, which is the percentage of total density to be captured for selecting threshold η, to observe its impact on the performance. Fig. 5.14 plots accuracy of MTS-MS at e = 30%, e = 50%, e = 70%, and e = 100% in terms of center location error (CLE), percentage of correctly tracked frames based on Pascal score (CDR), and precision at a fixed threshold of 20 pixels (Precision). It can be seen that according to all three evaluation metrics, MTS-MS achieves higher accuracy at smaller values (30% and 50%) of e in comparison to the larger values (70% and 100%) of e. For instance, MTS-MS at e = 30%, and e = 50% outperformed MTS-MS at e =

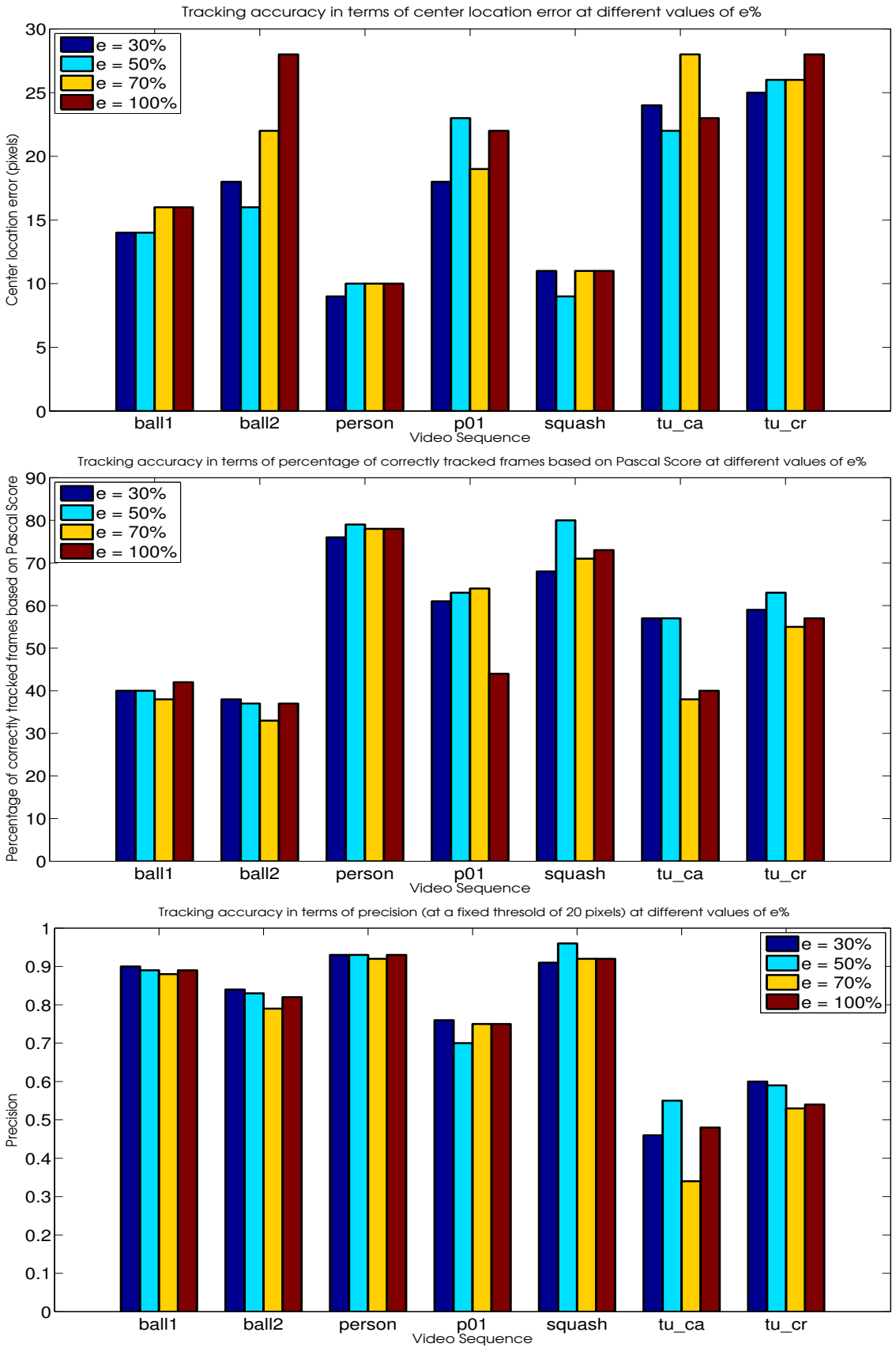


FIGURE 5.14: Performance of MTS-MS upon varying e values in terms of CLE, CDR, and Precision.

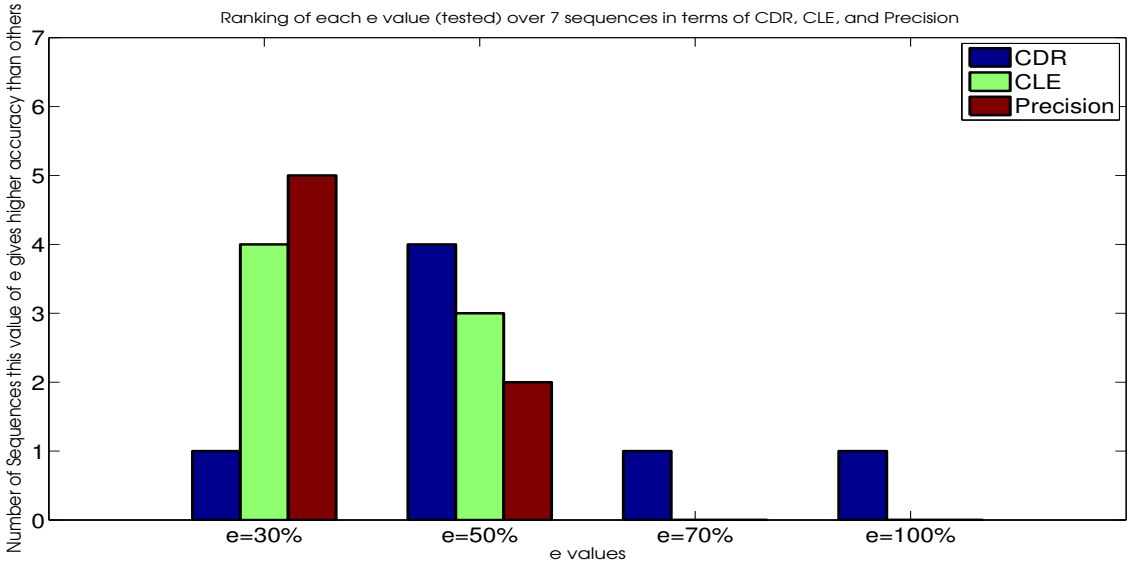


FIGURE 5.15: Performance summary of each e value in terms of CDR, CLE, and Precision. Each bar in this figure shows the number of sequences out of total (7) sequences the corresponding e value achieved higher accuracy than the other e values.

70%, and e = 100% in 5 out of 7 sequences in terms of CDR, and in all 7 sequences in terms of CLE and precision as displayed in Fig. 5.15. This might be due to the fact that large e values allow more state predictions and so increases search space. In such cases, the search method struggles to, or may not, reach global optimum with limited number of particles. In brief, it might be true that not all the predictions are equally accurate. Therefore, some of them should be deselected before defining cells in order to make search more accurate and efficient.

5.2.4 Discussion

MTS-MS displayed encouraging results, but it is important to understand why it did not improve the performance of the proposed framework (MTS) substantially i.e. in comparison to MTS-GWL. Firstly, MTS-MS is not capable of resolving visual ambiguities when a target changes appearance, since a fixed, simple appearance model is used. In other words, the new selection criterion does not have a mechanism to verify the presence of true target in these circumstances. In fact, it relies on visual appearance score to deselect some of the models (predictions) in the first stage, which makes it more sensitive to visual ambiguities. Furthermore, it derives motion information, extracted via spatial clustering, in a naive manner that does not help in overcoming visual ambiguities. Secondly, MTS-MS cannot prevent search in far-away non-target regions while the target is occluded. This is because the new selection criterion gives very high priority to the appearance information. It has no other reliable cues e.g. motion to bring on a

par with the appearance information to improve in these cases. Although the capability to avoid distraction to non-target regions during occlusion is a difficult task to achieve in practice, it can substantially improve the accuracy of the proposed framework.

5.3 Conclusion

In the beginning of this chapter, a search method was generalized to estimate the best target state around the state predictions generated in the proposed framework (MTS). These state predictions are produced by motion models learned and applied over multiple temporal scales. The search is modelled by assigning a certain area in state space, called a cell, to each state prediction. To search for the best target state in these cells, WLMCMC sampling was generalized to cells of variable size and location. Experimental evaluation revealed that the proposed framework (MTS) based on generalized WLMCMC (MTS-GWL) performed slightly better than the proposed framework based on the extension of particle filters (MTS-PF).

The experimental evaluation of MTS-GWL also gave a pointer to the fact that the effectiveness of a search method in the context of MTS relies on the location of state predictions in the state space. To gain further insight into the proposed framework and as a first step towards overcoming drawbacks in MTS, the model selection problem in the context of MTS was defined, and a new motion model selection criterion was proposed.

The new model selection criterion proceeds through two stages: rejection of state predictions with low visual likelihood scores, and spatial clustering. The rejection stage deselects some of the state predictions by automatically deciding a threshold since it believes that the surroundings of some predictions are not worth searching. The remaining predictions are spatially clustered to find possible groupings and remove redundancy. While establishing search regions, both appearance and motion information is utilized by assigning a cell to both each cluster center and to its highest weighted likelihood member.

Experiments were conducted to observe the impact of varying the e% parameter on the performance of MTS based on the new selection criterion (MTS-MS). e% is the percentage of the total density to be captured for selecting threshold η. Smaller values of e% revealed better results compared to larger values of e%. This suggests that it might be advantageous to reject some of the predictions before further processing (clustering and defining search regions) since not all of them are equally accurate.

The experimental evaluation of MTS-MS showed comparable results to MTS-GWL, but did not outperform MTS-GWL. MTS-MS is not robust to distractors while the target

is occluded and/or when it changes appearance. This is because it completely relies on appearance likelihood scores to filter predictions in the early stage of model selection. Furthermore, it lacks availability of some other cues e.g. motion to resolve appearance ambiguities while selecting models. Chapter 6 proposes some potential directions through which the aforementioned problems in MTS can be addressed.

Chapter 6

Conclusions and Future Work

The chapter begins by describing a summary of achievements and contributions made in this thesis (Section 6.1). Though the work presented in this dissertation provides a new perspective on key problems in visual tracking, there remains several limitations in the proposed approaches. Section 6.2 discusses these limitations and a few unaddressed topics. It further describes several possible ways through which they can be approached, and highlights potential future research directions.

6.1 Summary and Contributions

The thesis proposes and evaluates the idea of visual tracking over multiple temporal scales, and then further explores the proposed tracking framework. Three important questions arose from notion of visual tracking over multiple temporal scales. Chapters 3 and 4 address these questions and Chapter 5 explores some important components of the proposed tracking framework, resulting in important contributions to the field.

The three questions considered here are the following. (1) If a set of models, built over multiple temporal scales, is used, can better prediction performance be achieved compared to any single model from this set? (2) Is it possible to utilize the models generated from multiple time histories to predict possible target state multiple time-points ahead (in the future) to deal with occlusions? (3) Is it possible to select the most suitable model from the set of models and incorporate this into a tracking framework?

To address the first question, Chapter 3 introduced the practice of using ground truth data to evaluate the potential benefits of learning models over multiple temporal scales. In particular, experiments were performed assessing the usefulness of each of the models to predict motion and appearance in the ground truth data of several image sequences.

The study concluded that the prediction performance improves both, in case of motion and appearance, if a set of models, extracted from multiple temporal scales, is used along with a capability to select the most suitable one at each time-point. It was also observed that in the case of appearance prediction, this improvement in performance is not substantial when seen in comparison to motion prediction.

After observing the advantages of having a set of multi-scale models Chapter 4 attempted to answer questions 2, and 3. It proposed a visual tracker working over multiple temporal scales overcoming occlusions and abrupt motion variations. Motion models were learned from the target history at different temporal scales and applied over multiple temporal scales in the future. To estimate the posterior, the bootstrap particle filter was extended to propagate particles into the future at multiple temporal scales, guided by these motion models. Experiments were carried out to compare the performance of the proposed method with competing methods on both publicly available image sequences and some new sequences introduced in this thesis. The competing methods included best performing methods according to [Wu et al., 2013], and trackers capable of handling occlusions and abrupt motion variations. Results revealed that the proposed framework displayed better performance than the other methods in handling occlusions and rapid motion variations.

To understand the role of important components of the proposed tracking framework, and as an initial attempt towards further ameliorating its performance, Chapter 5 generalized a search method for multiple state predictions generated in multiple motion model frameworks, and proposed a new way of selecting motion models using their state predictions. First, a search method, which was originally proposed to search in a fixed grid of equal sized cells, was generalized to cells of variable size and location. The cells are formed around the predictions generated by motion models in the proposed framework. Experimental results showed that the proposed framework with the generalized search method performed somewhat better than the proposed framework based on the extension of particle filters. Although this improvement in performance is not substantial, this study has identified a new way of searching around multiple competing hypotheses in visual tracking. To investigate the role of model-selection in the context of the proposed framework and in the hope of further improving its performance, in the second half of Chapter 5 a novel motion model-selection technique was proposed. The proposed framework with this new model-selection technique did not outperform the original framework relying on the original existing model-set reduction method. However, this investigation exposed a few drawbacks, some of which this new model-selection technique could not address, in the proposed framework and their impact on the performance. Section 6.2 discusses those drawbacks in detail and underscores some relevant unaddressed topics.

6.2 Drawbacks, Unaddressed Topics and Future Work

Though the work described in this thesis has proposed a new framework for tackling two important problems in visual tracking and tried to further improve this framework, there remain some drawbacks and unaddressed topics. This section briefly underlines them and outlines several possible research directions through which they might be approached.

6.2.1 Distractors during Occlusion

The tracker proposed in Chapter 4 can jump to distractors while the target is occluded. Distractors are non-target regions in an image that are visually similar to the tracked target. During occlusions, the target observation is very weak or completely missing. Some off-target state prediction(s), which might be located close to a distractor, can lead the search to estimate the best target state upon or around this distractor. Off-target state predictions are located far away from the true target state. They are typically generated by motion models belonging to some previous time-point that is just a long way in time from the current time-point.

The aforementioned drawback can not only decrease tracking accuracy, but may introduce problems when the target re-appears after occlusion. Some or all of the state estimations during occlusions might be poor due to the drift of the tracker to non-target regions. Consequently, the motion models learned at time-points with poor state estimations would produce further off-target state predictions. These off-target state predictions may reduce the sampling efficiency of the search method. In the worst case scenario, the tracker can jump to distractors, when these off-target state predictions are located near distractors and the target changes appearance significantly.

One possible way to minimize the drift of the tracker away from the true target during occlusions is to improve the model-selection mechanism of MTS. If there were no state predictions lying close to distractors, it is quite unlikely that the search would be confused by the distractors. While choosing predictions when the image-based observation is weak or completely unavailable, the model-selection strategy, could be made to weight (trust) motion cues higher than appearance cues. The appearance cue is the visual likelihood score produced by the appearance model. Motion cues can be based on how reliable (close to the estimated target states) the state predictions from each motion model were over some time-period.

6.2.2 Revisiting Motion Model Selection

As discussed, it might be possible to enhance the performance of the proposed tracker by improving its motion model selection criterion. Looking back at the motion model selection scheme of chapter 4, Eq. 4.11 selects one motion model from each of the previous time-points by maximising visual likelihood scores corresponding to the state predictions generated from a given previous time-point.

Target motion is subject to variability over time and it might be that the motion models generated during period of occlusion are poor. In these situations, it is imperative to deselect off-target state predictions as these could lie on or close to distractors. These off-target predictions are more likely to be generated from those previous time-points that are farthest in time from the current time-point. The selection criterion in chapter 4 just takes into account the visual likelihood scores of the state predictions while selecting models. This makes it vulnerable to false positives as there could be one or more non-target regions bearing visual similarity to the tracked target. It does not distinguish between the predictions produced from the recent and the farthest time-points. Consequently, it is prone to selecting off-target predictions nearby distractors when the visual likelihood scores of these off-target and the on-target predictions are not very different. On-target predictions lie very close to the true target state.

A simple way to improve the existing model-selection criterion (based on Eq. 4.11) is to incorporate a penalty term that increases with time[1]. This penalty term would be higher for the state predictions produced from the time-point farthest in time from the current time-point and vice versa. It is based on the assumption that the state predictions from recent previous time-points are usually more accurate than the ones from the farthest time-points. This is not a strict assumption for the reasons described in the previous paragraph. With this penalty term, the visual likelihood scores of the state predictions from the farthest time-points should be much higher and not just the same than the state predictions from the recent ones to increase their odds of getting selected. As a result, the chances of picking false-positives will be reduced.

One potential way forward to include this penalty term is to have it in the inverse relationship with the visual likelihood score. With this formulation, the score for the model selection will be based on the ratio of visual likelihood score to this penalty term instead of just the visual likelihood score.

[1]While doing this the constraint of selecting at least one motion model from each previous time-point might have to be relaxed

6.2.3 Long-term Occlusions

The proposed tracking framework (MTS-L) in Chapter 4 has shown robustness towards full occlusions of different time-periods. However, it is susceptible to failure when faced with very long-terms occlusions (>40 frames). The key reason behind this inability to recover after long-term occlusions might be the unreliability of motion information. The longer the prediction-scales over which the motion models have to predict, the less likely it becomes that the true motion of the target will evolve according to the motion models. Another reason for failure might be that at very large prediction-scales, the spread of the (few) particles would be so large that they can completely miss the true mode of the target distribution.

A potential direction of research may be to investigate longer model-scales for learning motion models than the ones currently used. Their predictions over large prediction-scales might be more reliable than the shorter-scale models under long-term occlusions.

6.2.4 Capturing Appearance Variations

The implementations examined here use a fixed appearance model, which is built from information contained in the first frame of a tracking sequence. It may not stay valid when the target appearance varies considerably over the course of the sequence. If there exists considerable difference between the target model and the actual appearance of the target, the search method may not output an optimal solution. Under these circumstances, the accuracy of predictions by the motion models might become lower and may affect tracking accuracy. So, making the proposed framework adaptive to appearance variations may further improve its performance.

One possible solution to this extension would be to maintain an appearance pool, such as those proposed by [Kwon and Lee, 2010],[Park et al., 2012] while tracking. This appearance pool typically comprises target templates, and potentially covers possible aspects of the target appearance in the tracking sequence.

To track a frame, the following procedure will be adopted. Each template from this pool will be used by the search method to estimate the best target state corresponding to each template. To select the tracking result from these best target states, the deformation cost [Hong and Han, 2014] between the image observation associated with each best target state and the template (from the pool) corresponding to this best target state will be computed. The state with the minimum cost will be the tracking result at this frame.

6.2.5 Revisiting Poorly Tracked Frames

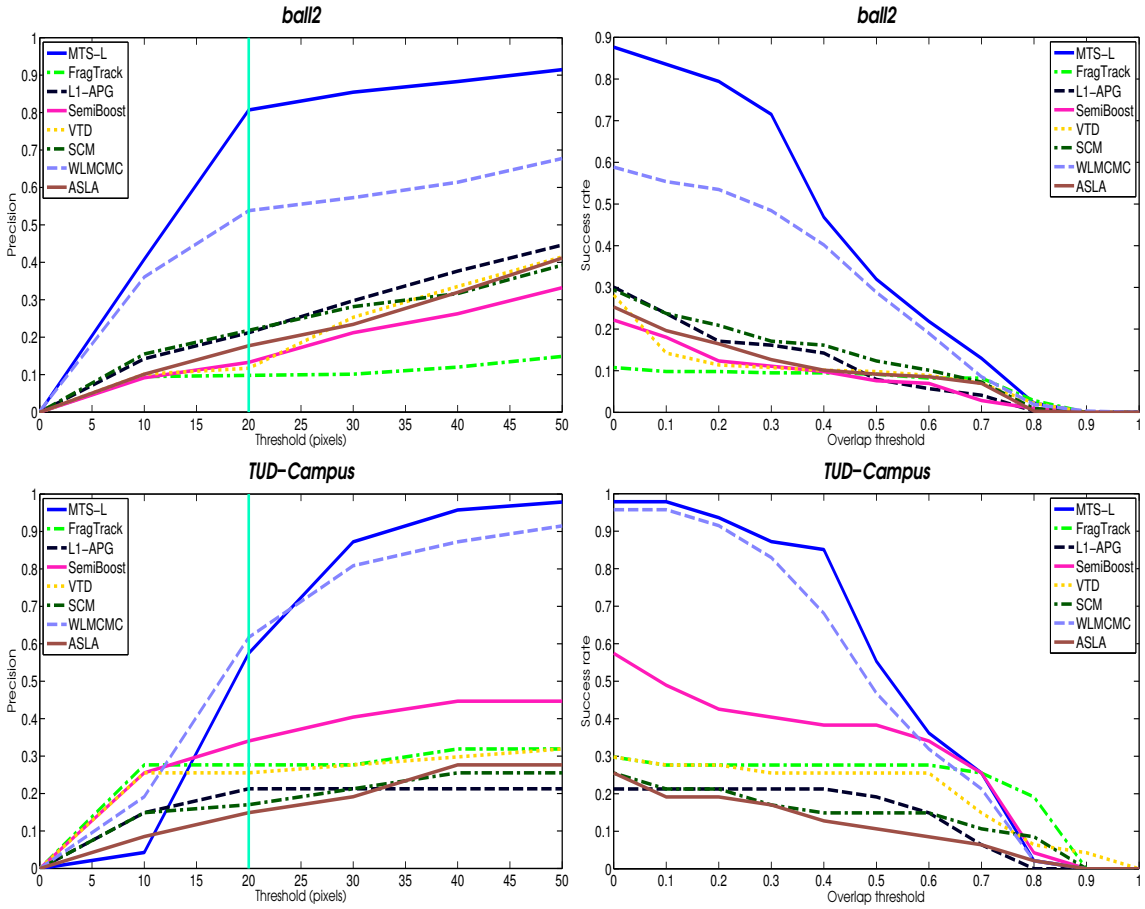
While tracking with the proposed framework (MTS), there would be some frames which have not been tracked properly. In other words, these are frames in which the estimated target state is far from the true target state. This might be due to the motion and/or appearance model used to track such frames being considerably different from the optimal models required to track these frames. The motion models learned from these frames can be erroneous, causing propagation of large tracking errors to other frames. Moreover, the tracking results of these frames would lower the overall tracking accuracy in a given image sequence. Re-visiting these frames for re-tracking once you have access to the video information beyond these frames might result in their proper tracking.

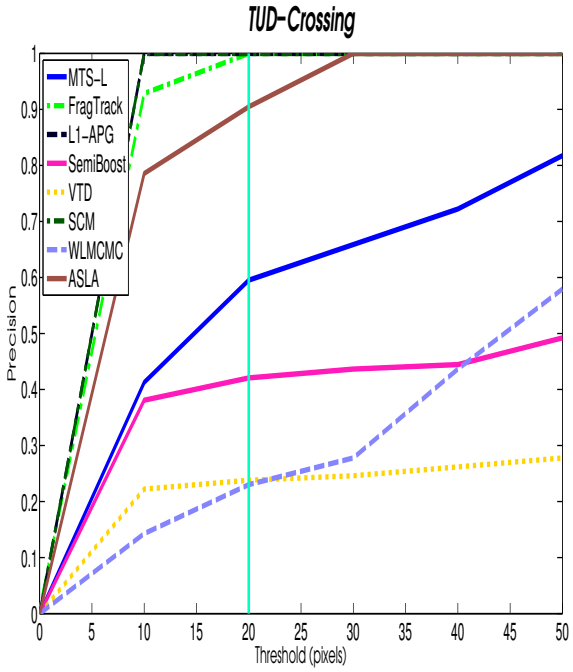
One potential way to minimize the impact of poorly tracked frames is to isolate and then re-track them. In other words, the motion information from these frames will be down-weighted when it is propagated to subsequent frames, and such frames will be re-visited later after they have been tracked for the first time.

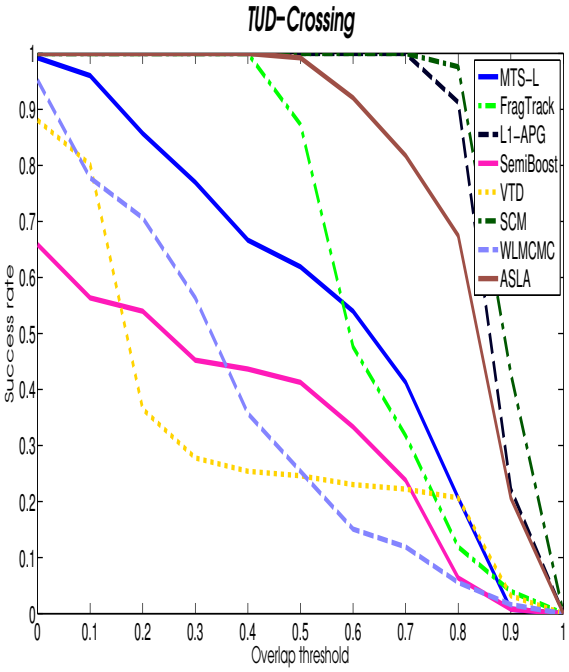
A simple method to identify poorly tracked frames might be following. Quantify how much the tracking result in a frame is similar to the previous appearances of the target. This score will then be used to down-weight the motion information from such frames. Poorly tracked frames having scores below than a pre-specified or dynamically chosen threshold will be nominated for re-tracking. PatchMatching as used by [Hong et al., 2013],[Hong and Han, 2014] can be employed in the first attempt to re-track nominated frames.

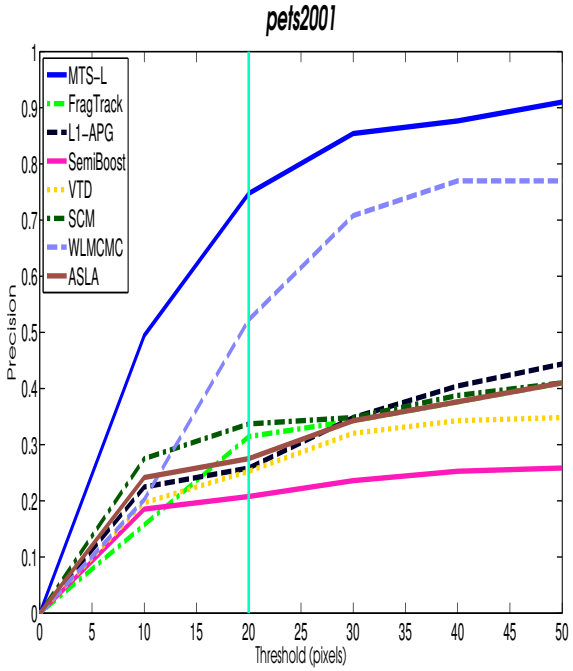
Appendix A

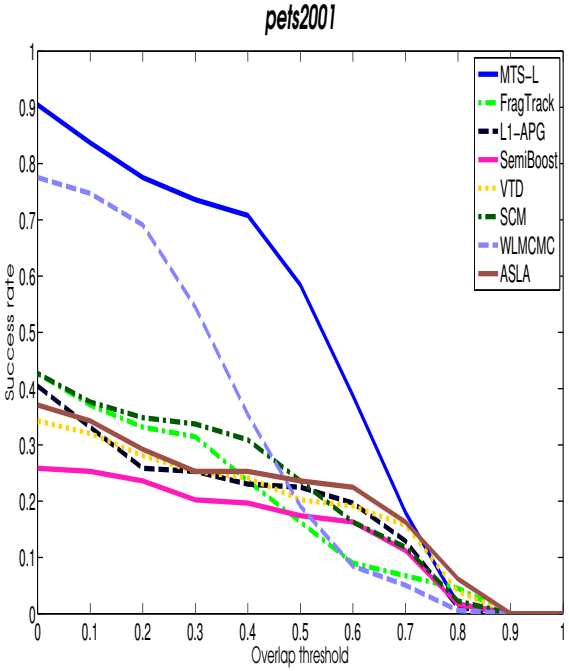
Quantitative Plots and Parameter List

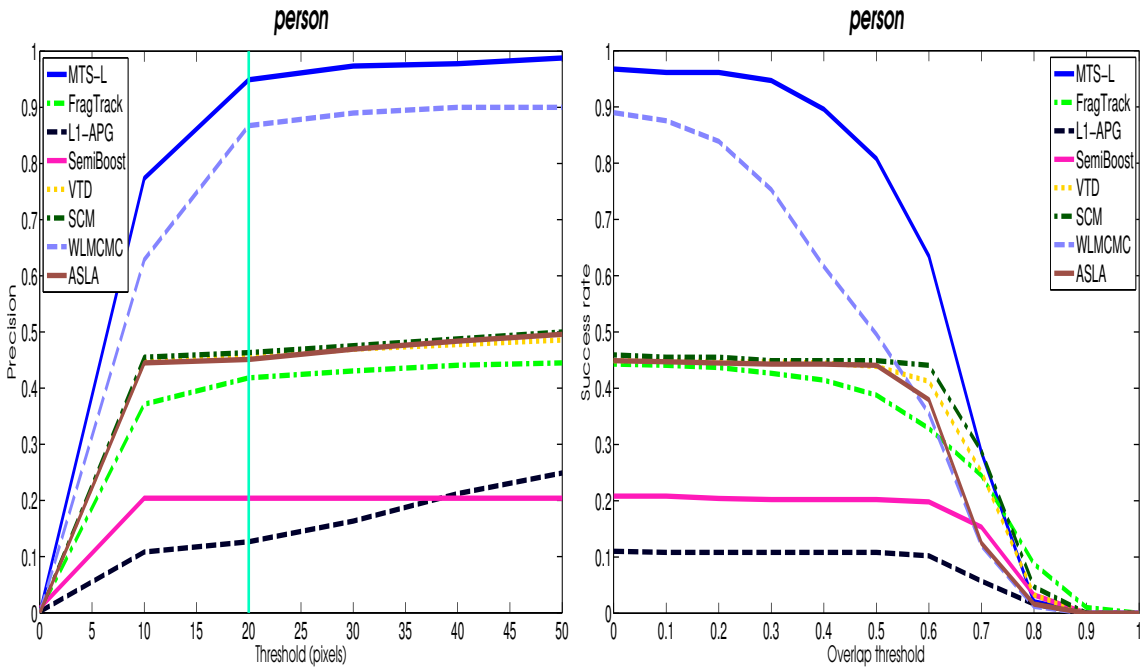


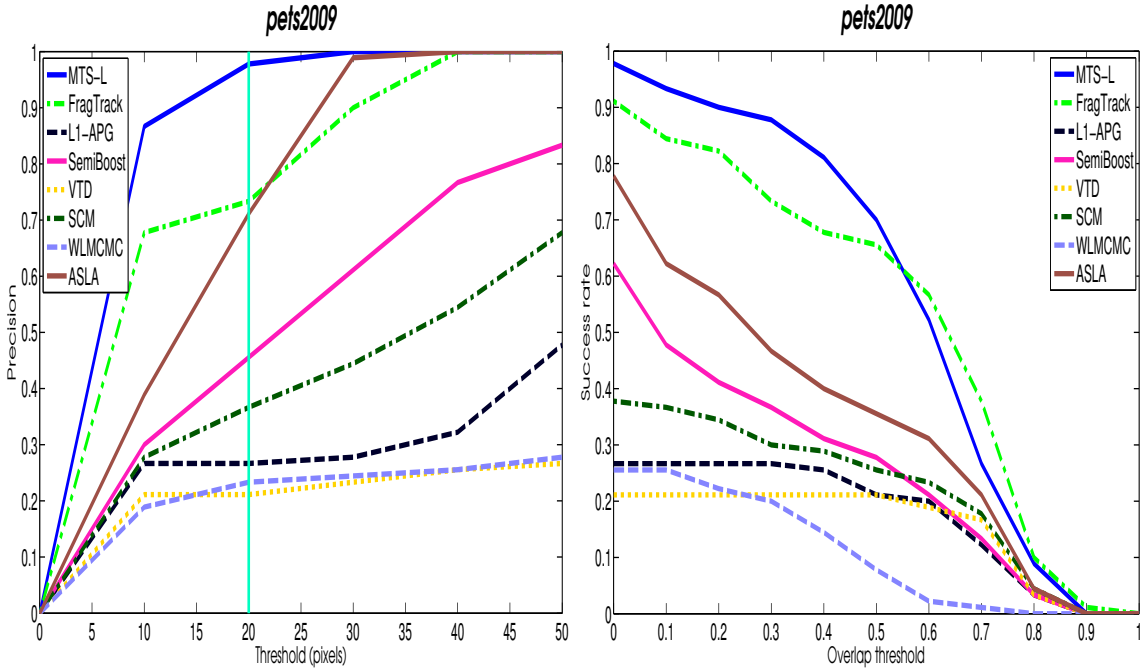

TUD-Crossing
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)

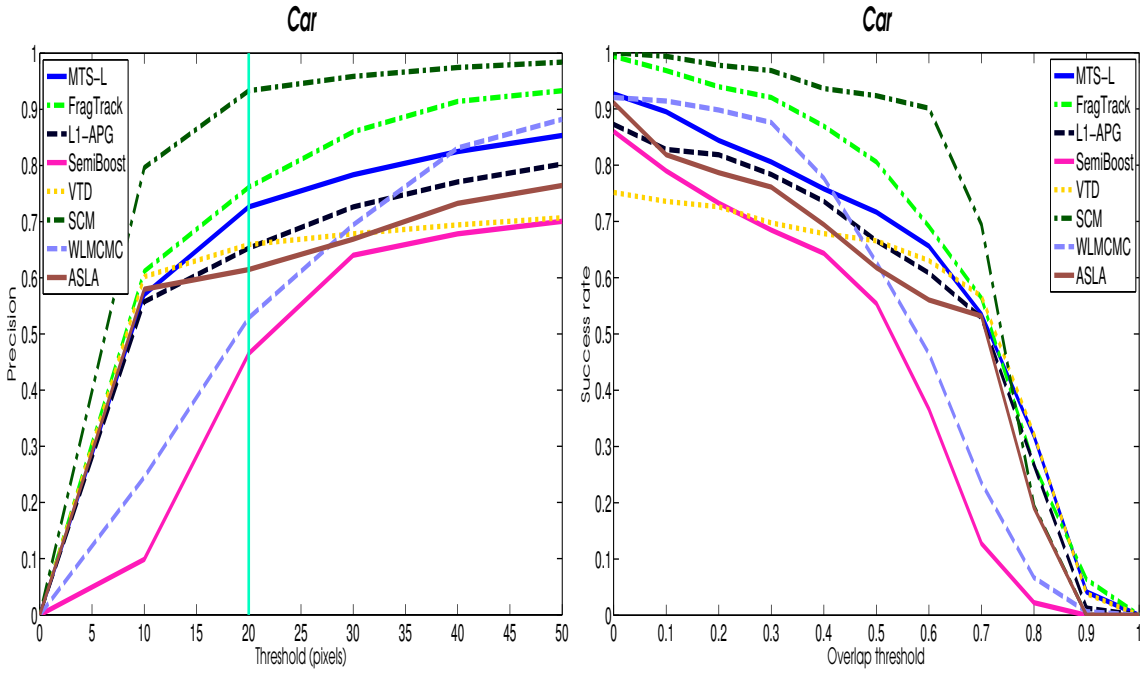

TUD-Crossing
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Success rate
Overlap threshold


pets2001
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)

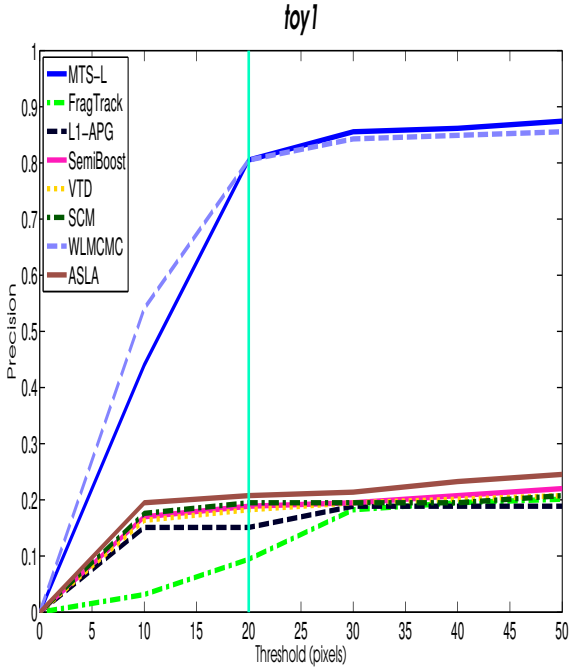

pets2001
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Success rate
Overlap threshold

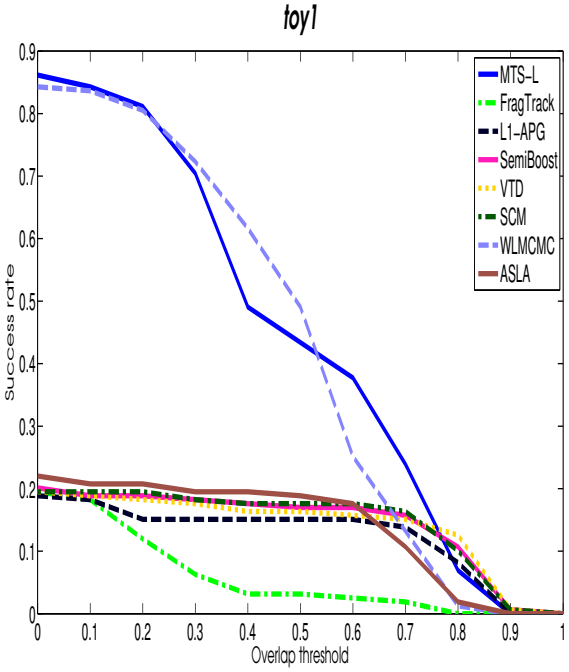

person
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)
person
Success rate
Overlap threshold

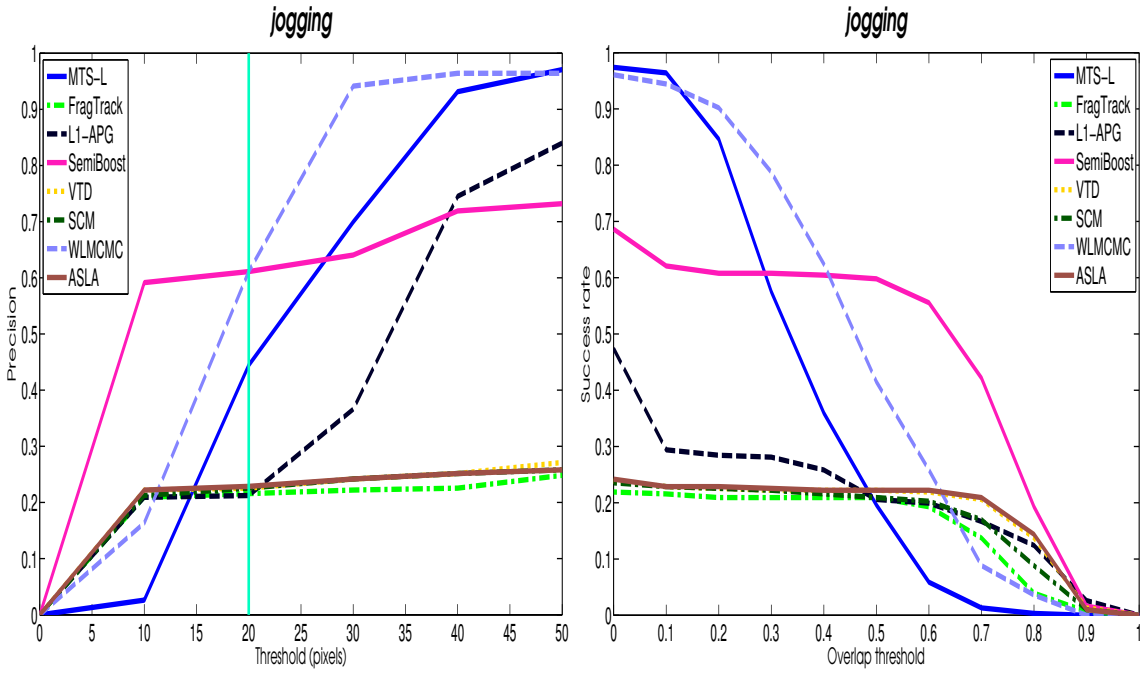

pets2009
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)
pets2009
Success rate
Overlap threshold

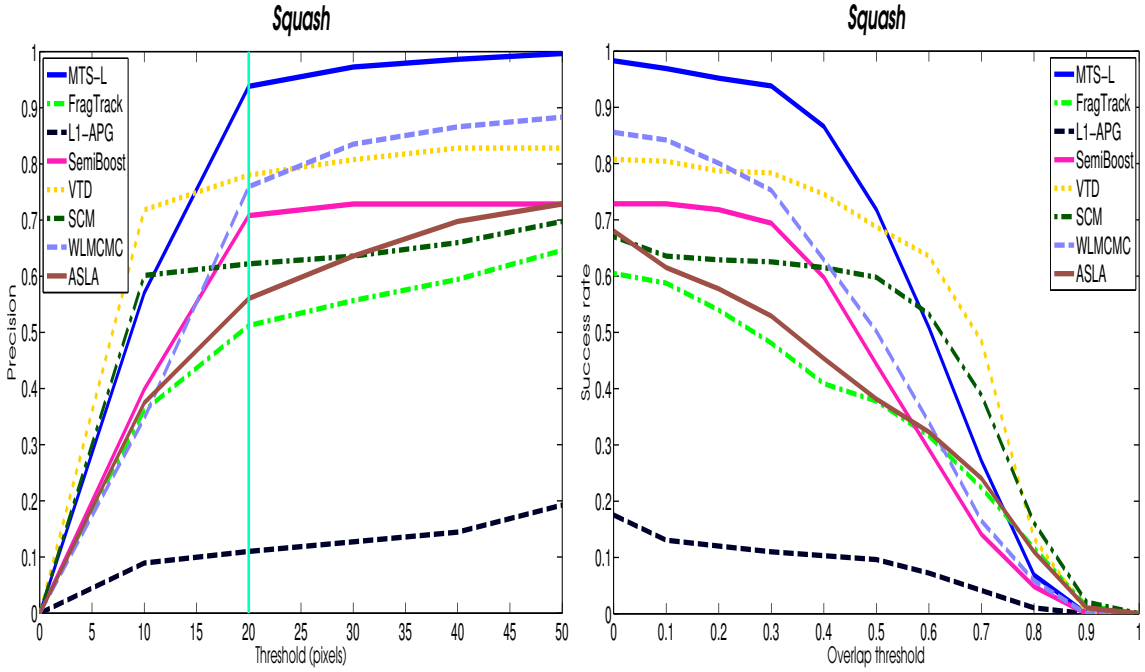

Car
Precision
Threshold (pixels)
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA

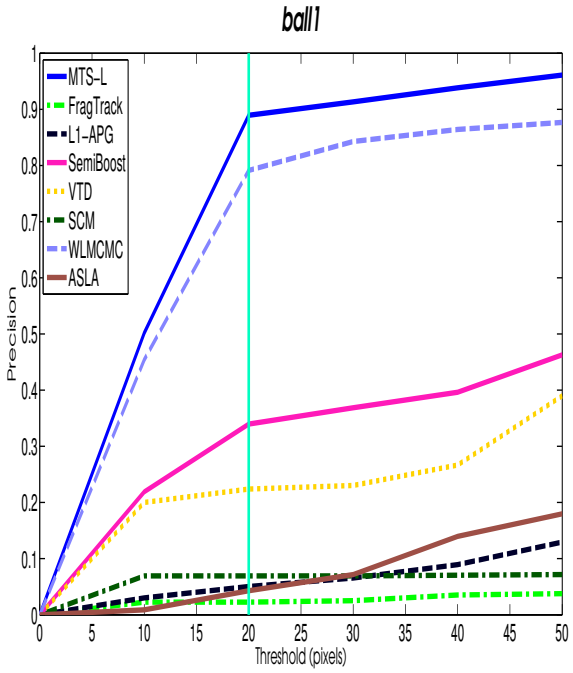
Car
Success rate
Overlap threshold
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA

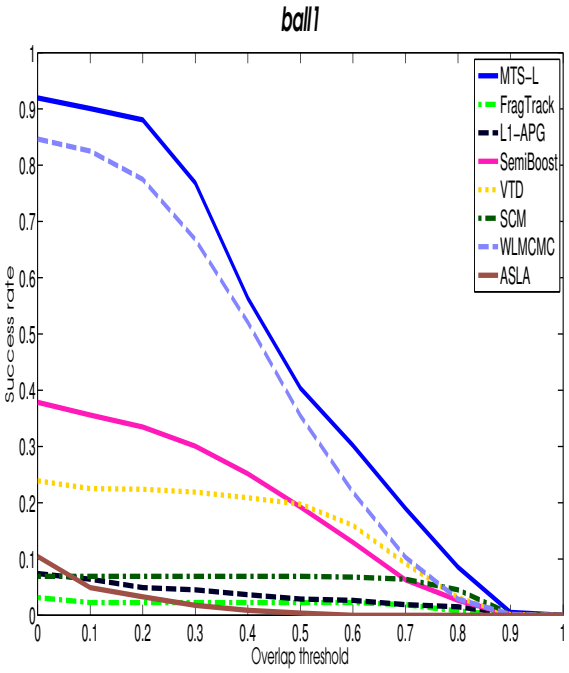

toy1
Precision
Threshold (pixels)
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA

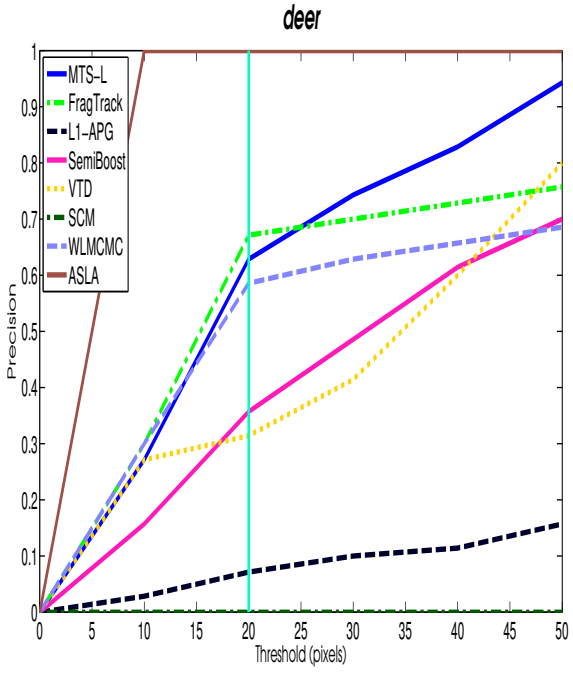

toy1
Success rate
Overlap threshold
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA

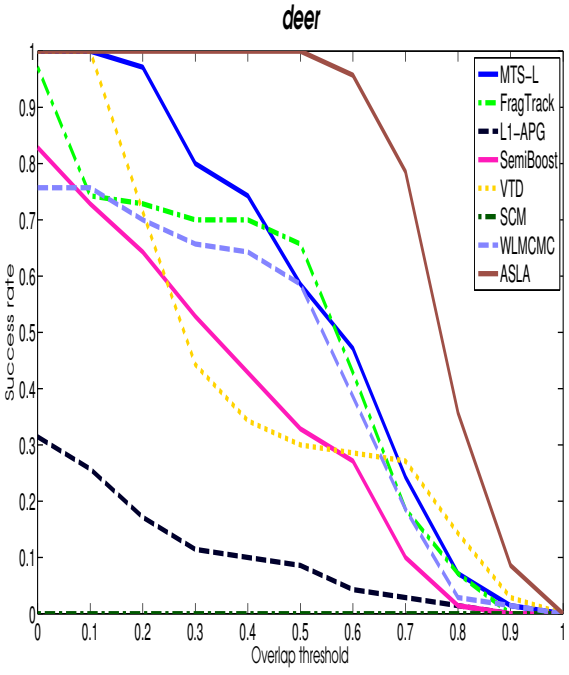

jogging
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)
jogging
Success rate
Overlap threshold


Squash
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)
Squash
Success rate
Overlap threshold


ball1
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)


ball1
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Success rate
Overlap threshold


deer
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Precision
Threshold (pixels)


deer
MTS-L
FragTrack
L1-APG
SemiBoost
VTD
SCM
WLMCMC
ASLA
Success rate
Overlap threshold

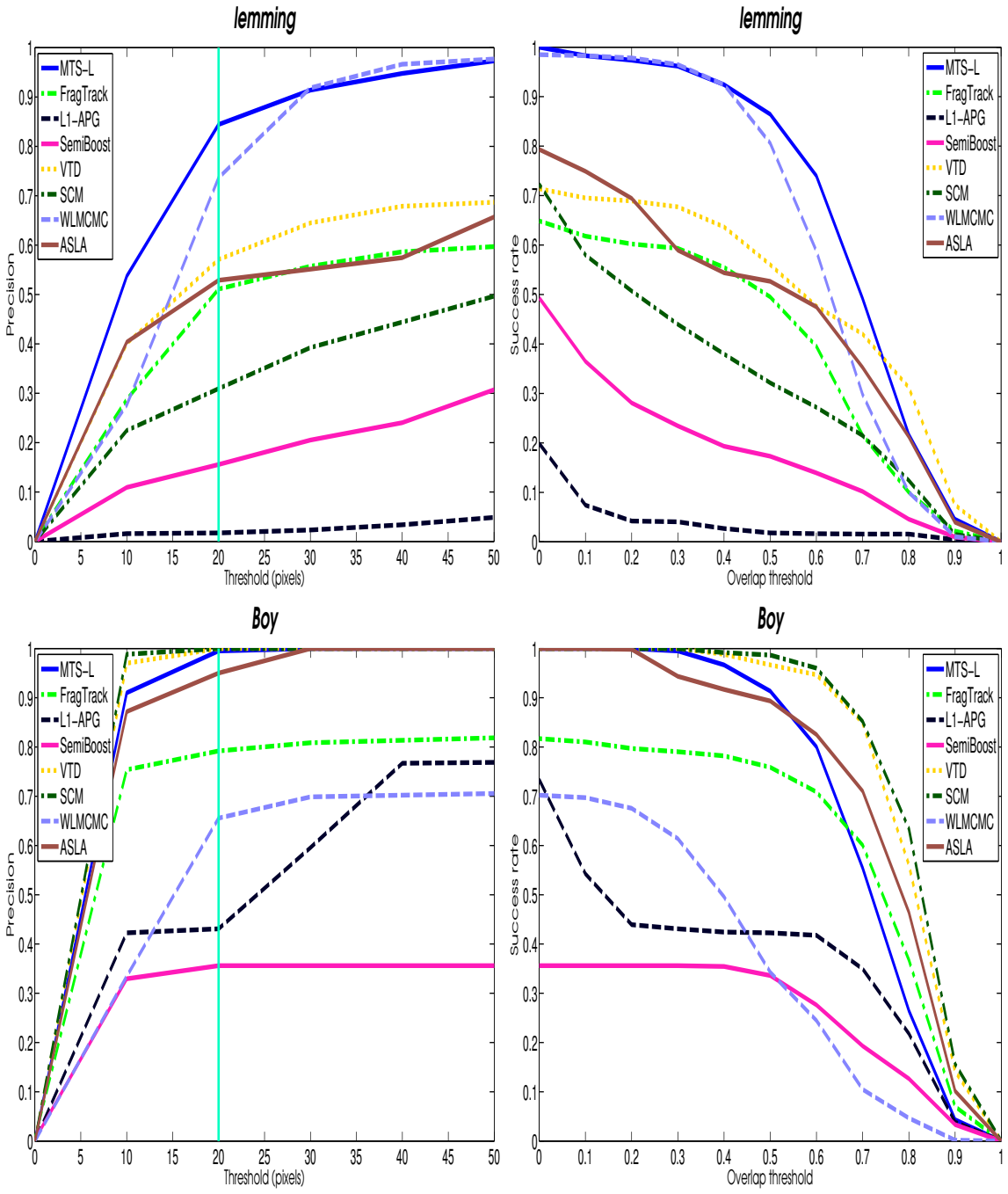


FIGURE A.1: **Precision and success plots for MTS-L, FragT, L1-APG, SemiBoost, VTD, SCM, WLMCMC, and ASLA in fourteen sequences used in chapter 4.**

TABLE A.1: **Parameters of different trackers used in the experiments.** $\sigma_{xy(v)}$ is the variance of the noise acting on the target's velocity in x and y directions. $\sigma_{xy(p)}$ denotes the standard deviation (in pixels) of the noise acting on the target's position in x and y directions. σ_e is the expected distance (in pixels) that a target covers between two time-steps. σ_1^x and σ_1^y are the standard deviations (in pixels) of the noise in the first motion model used in VTD. s_f is the search factor, which determines the size of the local search region with respect to the tracked target, and s_r is the search radius (pixels) around the target's position in previous time-step. n_p is the number of particles used in T_{NCV}, T_{RW}, and T_{TS}. T denotes the prediction-scales, and N is the number of particles propagated from time $t-k$ to time t in our proposed method. N is fixed at 20, and N_t is the total number of particles accumulated at time t.

Sequence	T_{NCV}	T_{RW}	T_{TS}		L1-APG	VTD	Semi	FragT	SCM,ASLA	MTS-(L,GWL,MS)	MTS-TS		
	$\sigma_{xy(v)}$	$\sigma_{xy(p)}$	σ_e	n_p	$\sigma_{xy(p)}$	σ_1^x,σ_1^y	s_f	s_r	$\sigma_{xy(p)}$	$\sigma_{xy(p)}$	σ_e	T	$N_t = N \times T$
TUD-Camp	6	9	3	180	8	4,2.82	3	10	12	4	3	9	180
TUD-Cross	0.5	1	1	500	1	2,1.414	2	7	5,5	1	1	25	500
PETS 2001	0.5	3	1	640	2	1,1.414	2	7	7,7	1	1	32	640
ball2	2	2	1	640	2	2,1.414	2	7	7,7	2	1	32	640
Person	1	1	1	400	2	2,1.414	2	7	7,7	1	1	20	400
PETS 2009	2	5	2	280	8	4,2.82	3	7	7,7	3	2	14	280
toy1	2	2	2	600	3	2,1.414	2	7	7,7	2	2	30	600
car	3	5	3	400	5	6,4.24	3	7	8,8	3	3	20	400
jogging	2	9	2	400	7	4,2.82	3	10	7,7	3	2	20	400
squash	6	8	4	100	8	8,5.65	6	10	18,18	6	4	5	100
ball1	11	13	8	280	12	13,9.19	11	11	21,21	9	8	14	280
deer	27	31	21	700	25	12,8.48	21	31	25,25	21	21	1	700
lemming	8	13	6	400	9	4,2.82	6	11	9,9	9	6	1	400
boy	9	13	9	200	9	6,4.24	6	20	9,9	9	9	1	200

Bibliography

Adam, Amit, Rivlin, Ehud, and Shimshoni, Ilan. Robust fragments-based tracking using the integral histogram. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805. IEEE, 2006.

Agarwal, Ankur and Triggs, Bill. 3d human pose from silhouettes by relevance vector regression. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–882. IEEE, 2004.

Akaike, Hirotugu. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

Andriluka, Mykhaylo, Roth, Stefan, and Schiele, Bernt. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

Arnaud, Elise and Mémin, Etienne. Partial linear gaussian models for tracking in image sequences using sequential monte carlo methods. *International Journal of Computer Vision*, 74(1):75–102, 2007.

Arulampalam, M Sanjeev, Maskell, Simon, Gordon, Neil, and Clapp, Tim. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

Arvind Ganesh, John Wright Allen Y. Yang Zihan Zhou, Andrew Wagner and Ma, Yi. *Compressed Sensing: Theory and Applications*, chapter Face recognition by sparse representation, pages 515–538. Cambridge University Press, 2011.

Avidan, Shai. Support vector tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1064–1072, 2004.

Avidan, Shai. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, 2007.

Babenko, Boris, Yang, Ming-Hsuan, and Belongie, Serge. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009.

Babenko, Boris, Yang, Ming-Hsuan, and Belongie, Serge. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, 2011.

Bai, Yancheng and Tang, Ming. Robust tracking via weakly supervised ranking svm. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1854–1861. IEEE, 2012.

Bao, Chenglong, Wu, Yi, Ling, Haibin, and Ji, Hui. Real time robust l1 tracker using accelerated proximal gradient approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1830–1837. IEEE, 2012.

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

Black, Michael J and Jepson, Allan D. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

Blumer, Anselm, Ehrenfeucht, Andrzej, Haussler, David, and Warmuth, Manfred K. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

Buchanan, Aeron and Fitzgibbon, Andrew. Combining local and global motion models for feature point tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Burt, Peter J and Adelson, Edward H. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4):532–540, 1983.

Cai, Yizheng, de Freitas, Nando, and Little, James. Robust visual tracking for multiple targets. *Computer Vision–ECCV 2006*, pages 107–118, 2006.

Cannons, Kevin. A review of visual tracking. 2008.

Canny, John. A variational approach to edge detection. In *The Third International Conference on Artificial Intelligence*, 1983.

Cappé, Olivier, Godsill, Simon J, and Moulines, Eric. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

Cehovin, Luka, Kristan, Matej, and Leonardis, Ales. An adaptive coupled-layer visual model for robust visual tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1363–1370. IEEE, 2011.

Chang, Cheng and Ansari, Rashid. Kernel particle filter for visual tracking. *Signal processing letters, IEEE*, 12(3):242–245, 2005.

Chen, Chih-Chang, Lin, Hsing-Hao, and Chen, Oscal T-C. Tracking and counting people in visual surveillance systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1425–1428. IEEE, 2011.

Chen, Li-Chih, Hsieh, Jun-Wei, Yan, Yilin, and Chen, Duan-Yu. Vehicle make and model recognition using sparse representation and symmetrical surfs. *Pattern Recognition*, 48(6):1979–1998, 2015.

Cifuentes, Cristina García, Sturzel, Marc, Jurie, Frédéric, Brostow, Gabriel J, et al. Motion models that only work sometimes. In *British Machive Vision Conference*, 2012.

Clymo, R.S. *Reporting research : a biologist's guide to articles, talks and posters*. Cambridge : Cambridge University Press, 2014.

Collins, Robert T, Liu, Yanxi, and Leordeanu, Marius. Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1631–1643, 2005.

Comaniciu, Dorin, Ramesh, Visvanathan, and Meer, Peter. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.

Comaniciu, Dorin, Ramesh, Visvanathan, and Meer, Peter. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

Dellaert, Frank, Burgard, Wolfram, Fox, Dieter, and Thrun, Sebastian. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

Dihl, Leandro, Jung, Cláudio Rosito, and Bins, José. Robust adaptive patch-based object tracking using weighted vector median filters. In *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*, pages 149–156. IEEE, 2011.

Dinh, Thang Ba, Vo, Nam, and Medioni, Gérard. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1177–1184. IEEE, 2011.

Dockstader, Shiloh L and Tekalp, A Murat. Multiple camera tracking of interacting and occluded human motion. *Proceedings of the IEEE*, 89(10):1441–1455, 2001.

Everingham, Mark, Van Gool, Luc, Williams, Christopher KI, Winn, John, and Zisserman, Andrew. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Fan, Jialue, Wu, Ying, and Dai, Shengyang. Discriminative spatial attention for robust tracking. In *Computer Vision–ECCV 2010*, pages 480–493. Springer, 2010.

Fei-Fei, Li, Fergus, Rob, and Perona, Pietro. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

Felzenszwalb, Pedro F, Girshick, Ross B, McAllester, David, and Ramanan, Deva. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

Fergus, Robert, Perona, Pietro, and Zisserman, Andrew. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2003.

Fleuret, Francois, Berclaz, Jerome, Lengagne, Richard, and Fua, Pascal. Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):267–282, 2008.

Fortmann, Thomas E, Bar-Shalom, Yaakov, and Scheffe, Molly. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173–184, 1983.

Fukunaga, Keinosuke and Hostetler, Larry. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

Furukawa, Yasutaka and Ponce, Jean. Carved visual hulls for image-based modeling. *International journal of computer vision*, 81(1):53–67, 2009.

Grabner, Helmut and Bischof, Horst. On-line boosting and vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 260–267. IEEE, 2006.

Grabner, Helmut, Leistner, Christian, and Bischof, Horst. Semi-supervised on-line boosting for robust tracking. In *Computer Vision–ECCV 2008*, pages 234–247. Springer, 2008.

Grabner, Helmut, Matas, Jiri, Van Gool, Luc, and Cattin, Philippe. Tracking the invisible: Learning where the object might be. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1285–1292. IEEE, 2010.

Grossberg, Stephen. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1):23–63, 1987.

Hampapur, Arun, Brown, Lisa, Connell, Jonathan, Ekin, Ahmet, Haas, Norman, Lu, Max, Merkl, Hans, and Pankanti, Sharath. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE*, 22 (2):38–51, 2005.

Han, Bohyung and Davis, Larry. On-line density-based appearance modeling for object tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1492–1499. IEEE, 2005.

Han, Bohyung and Davis, Larry S. Probabilistic fusion-based parameter estimation for visual tracking. *Computer Vision and Image Understanding*, 113(4):435–445, 2009.

Hare, Sam, Saffari, Amir, and Torr, Philip HS. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011.

Hastings, W Keith. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

He, Shengfeng, Yang, Qingxiong, Lau, Rynson W.H., Wang, Jiang, and Yang, Ming-Hsuan. Visual tracking via locality sensitive histograms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2427–2434, 2013.

Hong, Seunghoon and Han, Bohyung. Visual tracking by sampling tree-structured graphical models. In *Computer Vision–ECCV 2014*, pages 1–16. Springer, 2014.

Hong, Seunghoon, Kwak, Suha, and Han, Bohyung. Orderless tracking through model-averaged posterior estimation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2296–2303. IEEE, 2013.

Hsieh, Jun-Wei, Yu, Shih-Hao, Chen, Yung-Sheng, and Hu, Wen-Fong. Automatic traffic surveillance system for vehicle tracking and classification. *Intelligent Transportation Systems, IEEE Transactions on*, 7(2):175–187, 2006.

Hua, Gang and Wu, Ying. Multi-scale visual tracking by sequential belief propagation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–826. IEEE, 2004.

Huang, Chang, Li, Yuan, and Nevatia, Ramakant. Multiple target tracking by learning-based hierarchical association of detection responses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):898–910, 2013.

Huttenlocher, Daniel P, Noh, Jae J, and Rucklidge, William J. Tracking non-rigid objects in complex scenes. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 93–101. IEEE, 1993.

Isard, Michael and Blake, Andrew. Condensationconditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998a.

Isard, Michael and Blake, Andrew. A mixed-state condensation tracker with automatic model-switching. In *Computer Vision, 1998. Sixth International Conference on*, pages 107–112. IEEE, 1998b.

Jepson, Allan D, Fleet, David J, and El-Maraghi, Thomas F. Robust online appearance models for visual tracking. In *IEEE Conference on Computer Vision and and Pattern Recognition*, volume 1, pages 415–422, 2001.

Jia, Xu, Lu, Huchuan, and Yang, Ming-Hsuan. Visual tracking via adaptive structural local sparse appearance model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1822–1829. IEEE, 2012.

Johnson, Neil and Hogg, David. Learning the distribution of object trajectories for event recognition. *Image and vision computing*, 14(8):609–615, 1996.

Kalal, Zdenak. *TLD: TRACKING LEARNING DETECTION*. PhD thesis, 2011.

Kalal, Zdenek, Mikolajczyk, Krystian, and Matas, Jiri. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, 2012.

Kalman, Rudolph Emil. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

Karavasilis, Vasileios, Nikou, Christophoros, and Likas, Aristidis. Visual tracking using the earth mover's distance between gaussian mixtures and kalman filtering. *Image and Vision Computing*, 29(5):295–305, 2011.

Kass, Robert E and Raftery, Adrian E. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

Kaucic, Robert, Brooksby, Glen, Kaufhold, John, Hoogs, Anthony, et al. A unified framework for tracking through occlusions and across sensor gaps. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 990–997. IEEE, 2005.

Khan, Zia, Balch, Tucker, and Dellaert, Frank. Mcmc-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1805–1819, 2005.

Koenderink, Jan J. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

Koller-Meier, Esther B and Ade, Frank. Tracking multiple objects using the condensation algorithm. *Robotics and Autonomous Systems*, 34(2):93–105, 2001.

Korman, Simon and Avidan, Shai. Coherency sensitive hashing. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1607–1614. IEEE, 2011.

Kristan, Matej, Kovacic, Stanislav, Leonardis, Ales, and Pers, Janez. A two-stage dynamic model for visual tracking. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(6):1505–1520, 2010.

Kwak, Suha, Nam, Woonhyun, Han, Bohyung, and Han, Joon Hee. Learning occlusion with likelihoods for visual tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1551–1558. IEEE, 2011.

Kwon, Junseok and Lee, Kyoung Mu. Tracking of abrupt motion using wang-landau monte carlo estimation. In *Computer Vision–ECCV 2008*, pages 387–400. Springer, 2008.

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

Kwon, Junseok and Lee, Kyoung Mu. Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276. IEEE, 2010.

Kwon, Junseok and Lee, Kyoung Mu. Tracking by sampling trackers. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1195–1202. IEEE, 2011.

Kwon, Junseok and Lee, Kyoung Mu. Minimum uncertainty gap for robust visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2355–2362. IEEE, 2013.

Ladikos, Alexander, Benhimane, Selim, and Navab, Nassir. Multi-view reconstruction using narrow-band graph-cuts and surface normal optimization. In *BMVC*, pages 1–10, 2008.

Lazebnik, Svetlana, Schmid, Cordelia, and Ponce, Jean. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

Lerdsudwichai, Charay, Abdel-Mottaleb, Mohamed, Ansari, A, et al. Tracking multiple people with recovery from partial and total occlusion. *Pattern Recognition*, 38(7): 1059–1070, 2005.

Li, Hanxi, Shen, Chunhua, and Shi, Qinfeng. Real-time visual tracking using compressive sensing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1305–1312. IEEE, 2011.

Li, Yuan, Ai, Haizhou, Yamashita, Takayoshi, Lao, Shihong, and Kawade, Masato. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1728–1740, 2008.

Lim, Hwasup, Camps, Octavia I, Sznaier, Mario, and Morariu, Vlad I. Dynamic appearance modeling for human tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 751–757. IEEE, 2006.

Lindeberg, Tony. *Scale-space theory in computer vision*. Springer, 1993.

Lindeberg, Tony. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

Lindeberg, Tony. *Scale-Space*. John Wiley & Sons, Inc., 2007. ISBN 9780470050118.

Lucas, Bruce D, Kanade, Takeo, et al. An iterative image registration technique with an application to stereo vision. 81:674–679, 1981.

Madigan, David and Raftery, Adrian E. Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

Madrigal, Francisco, Rivera, Mariano, and Hayet, Jean-Bernard. Learning and regularizing motion models for enhancing particle filter-based target tracking. In *Advances in Image and Video Technology*, pages 287–298. Springer, 2012.

Maggio, Emilio and Cavallaro, Andrea. Hybrid particle filter and mean shift tracker with adaptive transition model. In *ICASSP (2)*, pages 221–224, 2005.

Matthews, Iain, Ishikawa, Takahiro, Baker, Simon, et al. The template update problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):810–815, 2004.

Mei, Xue and Ling, Haibin. Robust visual tracking using l1 minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443. IEEE, 2009.

Mei, Xue, Ling, Haibin, Wu, Yi, Blasch, Erik, and Bai, Li. Minimum error bounded efficient ? 1 tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1257–1264. IEEE, 2011.

Metropolis, Nicholas, Rosenbluth, Arianna W, Rosenbluth, Marshall N, Teller, Augusta H, and Teller, Edward. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Mikami, Dan, Otsuka, Kazuhiro, and Yamato, Junji. Memory-based particle filter for face pose tracking robust under complex dynamics. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 999–1006. IEEE, 2009.

Moreno-Noguer, Francesc, Sanfeliu, Alberto, and Samaras, Dimitris. Dependent multiple cue integration for robust tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(4):670–685, 2008.

Morris, Brendan Tran and Trivedi, Mohan M. Learning, modeling, and classification of vehicle track patterns from live video. *Intelligent Transportation Systems, IEEE Transactions on*, 9(3):425–437, 2008.

Murray, Don and Basu, Anup. Motion tracking with an active camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):449–459, 1994.

Naeem, Asad, Pridmore, Tony P, and Mills, Steven. Managing particle spread via hybrid particle filter/kernel mean shift tracking. In *BMVC*, pages 1–10, 2007.

North, Ben, Blake, Andrew, Isard, Michael, and Rittscher, Jens. Learning and classification of complex dynamics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(9):1016–1034, 2000.

Okuma, Kenji, Taleghani, Ali, De Freitas, Nando, Little, James J, and Lowe, David G. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

Park, Dong Woo, Kwon, Junseok, and Lee, Kyoung Mu. Robust visual tracking using autoregressive hidden markov model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1964–1971. IEEE, 2012.

Pavlovic, Vladimir, Rehg, James M, and MacCormick, John. Learning switching linear models of human motion. In *NIPS*, pages 981–987. Citeseer, 2000.

Pearce, Greg and Pears, Nick. Automatic make and model recognition from frontal images of cars. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 373–378. IEEE, 2011.

Pérez, Patrick, Hue, Carine, Vermaak, Jaco, and Gangnet, Michel. Color-based probabilistic tracking. In *Computer visionECCV 2002*, pages 661–675. Springer, 2002.

Perez, Patrick, Vermaak, Jaco, and Blake, Andrew. Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513, 2004.

Pernkopf, Franz. Tracking of multiple targets using online learning for reference model adaptation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(6):1465–1475, 2008.

Poiesi, Fabio and Cavallaro, Andrea. Tracking multiple high-density homogeneous targets. *IEEE Trans. Circuits Syst. Video Techn.*, 25(4):623–637, 2015. doi: 10.1109/TCSVT.2014.2344509. URL http://dx.doi.org/10.1109/TCSVT.2014.2344509.

Poole, Alex and Ball, Linden J. Eye tracking in hci and usability research. *Encyclopedia of human computer interaction*, 1:211–219, 2006.

Prewitt, Judith MS. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970.

Ramanan, Deva, Forsyth, David A, and Zisserman, Andrew. Tracking people by learning their appearance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):65–81, 2007.

Reddy, B Srinivasa and Chatterji, Biswanath N. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing*, 5(8):1266–1271, 1996.

Reid, Donald B. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

Roberts, Gareth O and Rosenthal, Jeffrey S. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

Roberts, Gareth O, Gelman, Andrew, Gilks, Walter R, et al. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

Rosenthal, Jeffrey S et al. Optimal proposal distributions and adaptive mcmc. *Handbook of Markov Chain Monte Carlo*, pages 93–112, 2011.

Ross, David A, Lim, Jongwoo, Lin, Ruei-Sung, and Yang, Ming-Hsuan. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77 (1-3):125–141, 2008.

Rowley, Henry, Baluja, Shumeet, Kanade, Takeo, et al. Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1): 23–38, 1998.

Saleemi, Imran, Shafique, Khurram, and Shah, Mubarak. Probabilistic modeling of scene dynamics for applications in visual surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1472–1485, 2009.

Santner, Jakob, Leistner, Christian, Saffari, Amir, Pock, Thomas, and Bischof, Horst. PROST Parallel Robust Online Simple Tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, 2010.

Schwarz, Gideon et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

Schweitzer, Haim, Bell, JW, and Wu, Feng. Very fast template matching. In *Computer VisionECCV 2002*, pages 358–372. Springer, 2002.

Sevilla-Lara, Laura and Learned-Miller, Erik. Distribution fields for tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1910–1917. IEEE, 2012.

Shafique, Khurram and Shah, Mubarak. A noniterative greedy algorithm for multiframe point correspondence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):51–65, 2005.

Shahed Nejhum, SM, Ho, Jeffrey, and Yang, Ming-Hsuan. Visual tracking with histograms and articulating blocks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

Shakhnarovich, Gregory, Viola, Paul, and Darrell, Trevor. Fast pose estimation with parameter-sensitive hashing. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 750–757. IEEE, 2003.

Shan, Caifeng, Tan, Tieniu, and Wei, Yucheng. Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recognition*, 40(7):1958–1970, 2007.

Sherlock, Christopher. *Methodology for inference on the Markov modulated Poisson process and theory for optimal scaling of the random walk Metropolis.* PhD thesis, Lancaster University, 2006.

Shotton, Jamie, Girshick, Ross, Fitzgibbon, Andrew, Sharp, Toby, Cook, Mat, Finocchio, Mark, Moore, Richard, Kohli, Pushmeet, Criminisi, Antonio, Kipman, Alex, et al. Efficient human pose estimation from single depth images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2821–2840, 2013.

Shu, Guang, Dehghan, Afshin, Oreifej, Omar, Hand, Emily, and Shah, Mubarak. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1815–1821. IEEE, 2012.

Smith, Kevin, Gatica-Perez, Daniel, and Odobez, Jean-Marc. Using particles to track varying numbers of interacting people. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 962–969. IEEE, 2005.

Smith, Kevin, Ba, Sileye O, Gatica-Perez, Daniel, and Odobez, Jean-Marc. Tracking the multi person wandering visual focus of attention. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 265–272. ACM, 2006.

Smith, Kevin C. *Bayesian methods for visual multi-object tracking with applications to human activity recognition.* PhD thesis, 2007.

Srinivas, Chukka, Hoogs, Anthony, Brooksby, Glen, Hu, Wensheng, et al. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 666–673. IEEE, 2006.

Sudderth, Erik B, Mandel, Michael I, Freeman, William T, and Willsky, Alan S. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *Nips*, volume 17, pages 1369–1376, 2004.

Sullivan, Josephine, Blake, Andrew, Isard, Michael, and MacCormick, John. Object localization by bayesian correlation. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1068–1075. IEEE, 1999.

Torre, Vincent and Poggio, Tomaso A. On edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):147–163, 1986.

Vedaldi, Andrea, Gulshan, Varun, Varma, Manik, and Zisserman, Andrew. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 606–613. IEEE, 2009.

Veenman, Cor J, Reinders, Marcel JT, and Backer, Eric. Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(1):54–72, 2001.

Viola, Paul and Jones, Michael J. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

Wang, Fugao and Landau, David P. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10):2050, 2001.

Wang, Liang, Hu, Weiming, and Tan, Tieniu. Recent developments in human motion analysis. *Pattern recognition*, 36(3):585–601, 2003.

Wang, Shu, Lu, Huchuan, Yang, Fan, and Yang, Ming-Hsuan. Superpixel tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1323–1330. IEEE, 2011.

Wang, Shuo, Xiong, Xiaocao, Xu, Yan, Wang, Chao, Zhang, Weiwei, Dai, Xiaofeng, and Zhang, Dongmei. Face-tracking as an augmented input in video games: enhancing presence, role-playing and control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1097–1106. ACM, 2006.

Witkin, Andrew P. Scale-space filtering: A new approach to multi-scale description. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, volume 9, pages 150–153. IEEE, 1984.

Wu, Bo and Nevatia, Ram. Tracking of multiple, partially occluded humans based on static body part detection. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 951–958. IEEE, 2006.

Wu, Bo and Nevatia, Ram. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

Wu, Yi, Lim, Jongwoo, and Yang, Ming-Hsuan. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013.

Xing, Junliang, Ai, Haizhou, and Lao, Shihong. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses.

In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1200–1207. IEEE, 2009.

Xing, Junliang, Gao, Jin, Li, Bing, Hu, Weiming, and Yan, Shuicheng. Robust object tracking with online multi-lifespan dictionary learning. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 665–672. IEEE, 2013.

Xiong, Fei, Camps, Octavia I, and Sznaier, Mario. Dynamic context for tracking behind occlusions. In *Computer Vision–ECCV 2012*, pages 580–593. Springer, 2012.

Yang, Changjiang, Duraiswami, Ramani, and Davis, Larry. Efficient mean-shift tracking via a new similarity measure. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 176–183. IEEE, 2005.

Yang, Hanxuan, Shao, Ling, Zheng, Feng, Wang, Liang, and Song, Zhan. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

Yang, Ming. *Context-aware and Attentional Visual Object Tracking.* PhD thesis, 2008.

Yang, Ming, Lv, Fengjun, Xu, Wei, and Gong, Yihong. Detection driven adaptive multi-cue integration for multiple human tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1554–1561. IEEE, 2009a.

Yang, Ming, Wu, Ying, and Hua, Gang. Context-aware visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(7):1195–1209, 2009b.

Yang, Ming-Hsuan. Object recognition. In *Encyclopedia of Database Systems*, pages 1936–1939. Springer, 2009.

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

Yilmaz, Alper, Javed, Omar, and Shah, Mubarak. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

Yin, Zhaozheng and Collins, Robert T. Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

Yu, Qian, Dinh, Thang Ba, and Medioni, Gérard. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *Computer Vision–ECCV 2008*, pages 678–691. Springer, 2008.

Zamir, Amir Roshan, Dehghan, Afshin, and Shah, Mubarak. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*, pages 343–356. Springer, 2012.

Zhang, Bin, Hsu, Meichun, and Dayal, Umesh. K-harmonic means-a spatial clustering algorithm with boosting. In *Temporal, Spatial, and Spatio-Temporal Data Mining*, pages 31–45. Springer, 2001.

Zhang, Kaihua, Zhang, Lei, and Yang, Ming-Hsuan. Real-time compressive tracking. In *Computer Vision–ECCV 2012*, pages 864–877. Springer, 2012a.

Zhang, Lu and van der Maaten, Laurens. Preserving structure in model-free tracking. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 36(4), 2014.

Zhang, Shun, Wang, Jinjun, Wang, Zelun, Gong, Yihong, and Liu, Yuehu. Multi-target tracking by learning local-to-global trajectory models. *Pattern Recognition*, 48(2): 580–590, 2015.

Zhang, Tianzhu, Ghanem, Bernard, Liu, Si, and Ahuja, Narendra. Robust visual tracking via multi-task sparse learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2042–2049. IEEE, 2012b.

Zhong, Bineng, Yao, Hongxun, Chen, Sheng, Ji, Rongrong, Yuan, Xiaotong, Liu, Shaohui, and Gao, Wen. Visual tracking via weakly supervised learning from multiple imperfect oracles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1323–1330. IEEE, 2010.

Zhong, Wei, Lu, Huchuan, and Yang, Ming-Hsuan. Robust object tracking via sparsity-based collaborative model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845. IEEE, 2012.

Zhou, Chenggang and Bhatt, RN. Understanding and improving the wang-landau algorithm. *Physical Review E*, 72(2):025701, 2005.

Zhou, Jie, Gao, Dashan, and Zhang, David. Moving vehicle detection for automatic traffic monitoring. *Vehicular Technology, IEEE Transactions on*, 56(1):51–59, 2007.

Zhou, Shaohua Kevin, Chellappa, Rama, and Moghaddam, Baback. Visual tracking and recognition using appearance-adaptive models in particle filters. *Image Processing, IEEE Transactions on*, 13(11):1491–1506, 2004.

Zhou, Xiuzhuang, Lu, Yao, Lu, Jiwen, and Zhou, Jie. Abrupt motion tracking via intensively adaptive markov-chain monte carlo sampling. *Image Processing, IEEE Transactions on*, 21(2):789–801, 2012.

Zhu, Qiang, Yeh, Mei-Chen, Cheng, Kwang-Ting, and Avidan, Shai. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.

Zimmermann, Karel, Matas, Jiri, and Svoboda, Tomas. Tracking by an optimal sequence of linear predictors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):677–692, 2009.