VISUAL TRACKING: DETECTING AND MAPPING OCCLUSION AND CAMOUFLAGE USING PROCESS-BEHAVIOUR CHARTS

Tissa Chandesa, BSc.

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

July 2013

Abstract

Visual tracking aims to identify a target object in each frame of an image sequence. It presents an important scientific problem since the human visual system is capable of tracking moving objects in a wide variety of situations. Artificial visual tracking systems also find practical application in areas such as visual surveillance, robotics, biomedical image analysis, medicine and the media. However, automatic visual tracking algorithms suffer from two common problems: occlusion and camouflage. Occlusion arises when another object, usually with different features, comes between the camera and the target. Camouflage occurs when an object with similar features lies behind the target and makes the target invisible from the camera's point of view. Either of these disruptive events can cause a tracker to lose its target and fail.

This thesis focuses on the detection of occlusion and camouflage in a particle-filter based tracking algorithm. Particle filters are commonly used in tracking. Each particle represents a single hypothesis as to the target's state, with some probability of being correct. The collection of particles tracking a target in each frame of an image sequence is called a particle set. The configuration of that particle set provides vital information about the state of the tracker. The work detailed in this thesis presents three innovative approaches to detecting occlusion and/or camouflage during tracking by evaluating the fluctuating behaviours of the particle set and detecting anomalies using a graphical statistical tool called a process-behaviour chart. The information produced by the process-behaviour chart is then used to map out the boundary of the interfering object, providing valuable information about the viewed environment.

A method based on the medial axis of a novel representation of particle distribution termed the Particle History Image was found to perform best over a set of real and artificial test sequences, detecting 90% of occlusion and 100% of camouflage events. Key advantages of the method over previous work in the area are: (1) it is less sensitive to false data and less likely to fire prematurely; (2) it provides a better representation of particle set behaviour by aggregating particles over a longer time period and (3) the use of a training set to parameterise the process-behaviour charts means that comparisons are being made between measurements that are both made over extended time periods, improving reliability.

Published Work

Conference paper:

Chandesa, T., Pridmore, T. & Bargiela, A. (2009a) Published. Detecting occlusion and camouflage during visual tracking. *IEEE International Conference on Signal and Image Processing Applications (ICSIPA '09)*, November 18-19 2009a Kuala Lumpur, Malaysia. pp.468-473.

Poster presentation:

Chandesa, T., Pridmore, T. & Bargiela, A. (2009b) *Using Process-Behaviour charts to detect Occlusion and Camouflage in Visual Tracking*. Presented in: Research for a Better Tomorrow: Impacts in the 21st Centrury, University of Nottingham, Malaysia.

Acknowledgment

I am really lucky as throughout the duration of my PhD, I was always surrounded by people who constantly gave me unvarying support, guidance, motivation and encouragement. I believe that the accomplishment of my PhD is a testament of the positive affect of being in the company of all these great individuals. Therefore, I would like to take this opportunity to express my heartfelt gratitude to all these wonderful people.

I am forever grateful to my advisors, Professor Tony Pridmore and Professor Andrzej Bargiela for their continuous wisdom, guidance and support. Their rich knowledge, wealth of experience and passion for research make me feel blessed and fortunate to have them both as my mentors. Being their research student, I have learnt the true meaning of research and how to appreciate as well as excel in research. I believe that I am a better researcher because of them and I intend to put into practice all of the wisdom that they have taught and shared with me as I continue to build a career in research and academia in the near future.

I would like to say thank you from the bottom of my heart to my loving parents, especially to my mother, for all of her endless guidance, encouragement and support that she gave to me throughout my PhD journey.

I would also like to sincerely thank Professor Graham Kendall and Dr. Timothy Brailsford for their continuous support during the duration of my PhD. I am also thankful to all of the academic staffs, administrative staffs as well as the technical staffs at the school of Computer Science at the Malaysian Campus as well as at the Jubilee Campus for their continuous guidance and support. I would also like to thank everyone at the Faculty of Science, Graduate School as well as IT support at the Malaysian Campus for the constant support they gave to me.

Finally, I would like to convey my truthful appreciation to all of my amazing friends for their help and support. I’m truly thankful to them all for finding the time to take part and help me perform numerous research experiments that were carried out as part of my PhD research work.

Once more, thank you to all.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1. General introduction and motivation

Visual tracking is the process of generating inferences about the motion of an object or set of objects from a time-ordered image sequence. The relationships between features from neighbouring frames are analysed to recover motion. Data is provided only at selected targets, but both the camera and/or target can be in motion.

Tracking has received much attention and is a key problem in computer vision. The visual environment is naturally dynamic: people, animals and vehicles move almost continuously, providing constantly changing image data. Actual and potential applications are numerous and include tasks in visual surveillance, media analysis and generation, robotics, target tracking, biomedical image analysis and medicine.

Trackers commonly consist of three main components – an appearance model which describes the image feature associated with the target, a motion model which describes its likely movement between frames and a tracking engine which combines the two to recover target motion and achieve tracking. Tracking engines are mostly built based upon model estimation concepts, such as Kalman filter or Sequential Monte Carlo methods. Kalman filtering (Kalman, 1960), particle filtering (Isard et al., 1996) and mean-shift algorithms (Comaniciu et al., 2003, Comaniciu, 2003) are among the most widely used approaches.

Many powerful tracking techniques exist. Recent reviews on visual tracking have suggested that current tracking methods can be categorised in a number of different ways. Yilmaz et al (Yilmaz et al., 2006) use the representation of the tracked object to distinguish three categories: kernel tracking, silhouette tracking and point tracking. Babu et al (Babu et al., 2007) identify four broad types of tracker based on the tracking process used: gradient based methods, feature-matching approaches, knowledge-based tracking algorithms and learning-based approaches.

As tracking techniques and systems develop, evaluation becomes more important. Trackers are usually evaluated on the basis of some assessment of their accuracy, robustness and computational efficiency. Measures of accuracy reflect the precision with which the tracker output represents the target's motion. A tracker is considered to be robust when it remains associated with its target throughout the input image sequence. Computational efficiency is important in some applications (e.g. real-time surveillance or robotics), but is generally less important than metrics which assess the quality of tracking performed (Black et al., 2003, Ellis, 2002, Pound et al., 2007).

While many effective tracking algorithms exist common problems remain. Some can cause deterioration in the tracker's accuracy while others affect robustness. High levels of image noise can reduce robustness by disrupting the extraction of target features, but at moderate levels random variations in pixel values are more likely to reduce accuracy. Motion noise in the form of irregular target movement complicates tracking as motion models typically assume constant or smoothly varying movement. Again, highly irregular motion can reduce robustness, but in many cases motion noise affects only tracking accuracy. Variation in the target's illumination can arise from changes in the real or artificial sources lighting the scene, or in the target's relationship to those sources. Illumination changes can disrupt the way the target is represented in the image data, making it a poor match to the appearance model and reducing accuracy and sometimes robustness, depending on the appearance model used. Reduction of the target's visibility from the camera's point of view is likely to cause the tracker to fail. There are two ways visibility of the target can be disrupted: occlusion and/or camouflage. Occlusion happens when objects with different appearances to the target fall between the target and camera. Camouflage occurs when objects with similar appearances to the target form the background and distract the tracker away from the true target.

Occlusion and camouflage are important because of the severe effect they have on tracker robustness. The occurrence of occlusion and camouflage during tracking are common and, if the camera is fixed, inevitable in most situations. At present, many tracking algorithms are able to remain associated with their target(s) through a reasonable number (tens or hundreds) of frames. Occlusion and camouflage will, however, eventually cause tracking to fail and present tracking algorithms do not allow target(s) to be reacquired with any reliability. Though occlusion and camouflage events typically occupy only a small number of frames, the length of time over which tracking can be expected to be successful increases substantially with every event that the tracker can deal with. The starting point of the work reported here is that detailed analysis of occlusion and camouflage events is necessary and important if the scope and performance of current visual tracking algorithms is to be improved.

The work reported here focuses on the detection of these two disruptive events in a particle-filtered based tracking framework. Particle filters are commonly used in tracking. Each particle represents a single hypothesis as to the target's state, with some probability of being correct. The likely position of the target in each frame is represented by a collection of particles, referred to here as a particle set. Particle sets are of fixed size, but the distribution of the particles making up the set varies from image to image as the tracker works to maintain an accurate description of its target's location in each image. A key feature of particle sets is that they are free to form multi-modal distributions, simultaneously representing alternative, competing interpretations.

It should be stressed here that the detection of occlusion and/or camouflage is focused on target object(s) moving in 2D motion, where the changes to the target's position in each frame is taken into consideration and the camera remains stationary throughout.

1.2. Research aim

The aims of this thesis are:

1. To investigate the effects of interactions with interfering objects on the particle sets that represent estimates of target state in particle filter-based tracking algorithms.
2. To develop methods of detecting these events, specifically occlusion and camouflage of the tracked object.
3. To exploit the developed methods to create maps that outline areas of the viewed environment where occlusion and/or camouflage occur when tracking multiple targets moving in a static scene.

Although, occlusion and camouflage may only take a few frames to occur, trackers which can survive these events are likely to track their target(s) for much longer time periods. Present trackers do not explicitly detect and react to the occurrence of occlusion and camouflage. Instead alternative approaches are used to cope with the occurrence of occlusion and camouflage. These include keeping the tracker more tightly focused on the target by manipulating the motion and/or appearance models employed and/or the engine used to apply them (Yang et al., 2011, Babu et al., 2007, Yilmaz et al., 2006, Xiang, 2011) or identifying areas of the environment in which the target is more likely to appear.

The assumption underlying this thesis is that the inherent complexity of real tracking problems means that the danger of noise and background objects disrupting tracking will never entirely be removed. Rather than attempt to create trackers which are impervious to such hazards, the solution lies in detecting and reacting to the disruptive events that they cause.

1.3. Contributions

Following an examination of the fluctuations in particle set distribution caused by these events, three approaches to the detection of occlusion and camouflage are considered here:

1. Particle clustering and process-behaviour charts
    - Particle clustering is used within a particle-filtered tracker to analyse the effects of occlusion and camouflage and identify cues related to them.
    - The information gathered from that analysis is exploited by using a process-behaviour chart to allow the tracker to determine the occurrences of occlusion and camouflage by monitoring the control points.
2. Particle History Images (PHI), Particle Boundary Images (PBI) and process behaviour charts
    - A novel view-based representation of apparent motion is constructed from the particle sets employed by a particle filter-based tracking algorithm, introducing the notion of the Particle History Image (PHI).
    - Applying a texture edge detection algorithm to the PHI produces a Particle Boundary Image (PBI) which highlights the boundary of the particle spread in the PHI.
    - Information pertaining to the width of the particle spread is gathered by analysing the PBI.
    - The resulting information is exploited using a process-behaviour chart to allow the tracker to detect the occurrences of occlusion and/or camouflage by monitoring the process-behaviour chart control points.
3. Comparison of the tracker's estimate of the target's path and the medial axis of the PBI
    - Information pertaining to the alignment of the tracker's estimated path and the medial axis path is computed.

- The computed data is exploited using a process-behaviour chart to detect occurrences of occlusion and/or camouflage during tracking by monitoring the process-behaviour chart control points.

The effectiveness of each approach is evaluated when tracking multiple targets moving within a static scene. The resulting occlusion and camouflage events are then used to build Gaussian Mixture Model maps of the boundaries of the interfering objects, marking areas of the background environment in which occlusion and/or camouflage are likely to occur.

The rest of this thesis describes how these contributions were achieved.

1.4. Thesis overview

This thesis is structured as follows:

Chapter 2: Motivation and Background

An examination of the literature detailing how present trackers handle the occurrence of occlusion and/or camouflage is given. Kalman filtering and particle filtering (e.g. Condensation) model estimation concepts and their implementation are presented. A particle filter-based tracker is described and applied to test image sequences exhibiting occlusion and camouflage. The results gathered in relation to the particle set behaviour associated with these events are analysed.

Chapter 3: A Particle Clustering Approach

Particle clustering via an implementation of a Gaussian Mixture Model within a particle-filter based tracker is presented. Use of a process-behaviour chart to detect occlusion and camouflage using information gathered from particle clustering is detailed. The performance of the process-behaviour chart algorithm and the Condensation algorithm is evaluated.

Chapter 4: Measuring Particle Spread

Two novel representations: (1) Particle History Images (PHI) and (2) Particle Boundary Images (PBI) are presented and later analysed to measure the width of the particle spread. The result of using a process-behaviour chart to exploit the information produced via two approaches to detect occlusion and/or camouflage during tracking is detailed.

Chapter 5: Scene Mapping

The novel approaches presented in Chapter 3 and Chapter 4 are applied to scene mapping. The accuracy of each approach at detecting occlusion and/or camouflage when tracking multiple targets moving within a static scene is measured. Scene maps built as a result of exploiting the results produced from each approach using process-behaviour chart is analysed. Additionally, the accuracy of these scene maps at describing the interfering objects structures of the viewed environment are also evaluated.

Chapter 6: Contributions and Future Work

Contributions of this thesis are outlined and possible future work is presented.

Chapter 2 Motivation and Background

2.1. Aim

The aims of this chapter are:

1. To describe the processes of occlusion and camouflage and explain their relevance to visual tracking.
2. To review how present trackers handle the occurrence of occlusion and camouflage.
3. To review key concepts in visual tracking.
4. To apply a particle filter-based tracker to test image sequences exhibiting occlusion and camouflage and to consider the particle set behaviour associated with these events.

2.2. Motivation

2.2.1. Normal tracking, occlusion, self-occlusion and camouflage

Normal tracking occurs when a target is fully visible from the view of the camera and is successfully tracked from the start of the tracking process to the end (e.g. Figure 2.5).

Occlusion occurs when a target is lost from the view of the camera as an occluding object, usually with different features, falls between the camera and the target. Occlusion can be static, in which a fixed object occludes the target, or dynamic, in which another moving object comes between the target and camera. Figure 2.1 shows an example of the process of occlusion. The yellow circle is the target being tracked while the blue rectangle is the occluding object. The yellow circle is fully or partially visible for the first two frames in the image sequence but in the last frame; the blue rectangle replaces it in the foreground and makes the yellow circle invisible from the camera’s point of view. Self-occlusion, meanwhile, occurs when some part of the target occludes the feature(s) being tracked (e.g. Figure 2.16). The most common cause of self-occlusion is 3D rotation of the target, though articulated objects can self-occlude when one part of the target moves in front of another.

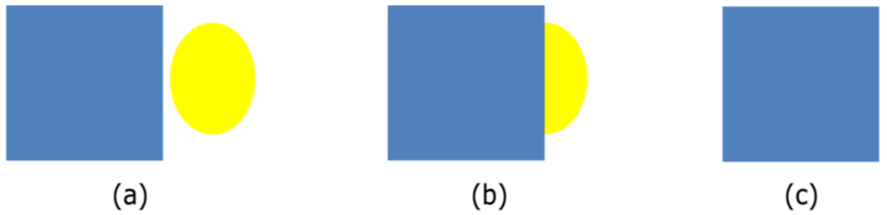


Figure 2.1 The process of occlusion.

Camouflage on the contrary occurs when a target becomes invisible due to the presence of a larger object with similar features in the background. While the target remains in full view, the similarity of their appearance makes the camouflaging and target objects indistinguishable. Figure 2.2 shows an example of the process of camouflage. The yellow circle is the target being tracked in this image sequence while the yellow rectangle is the camouflaging object. For the first two frames, the yellow circle is fully or partially distinguishable. However, in the last frame; the yellow circle is no longer visible, having become embedded in the yellow rectangular in the background.

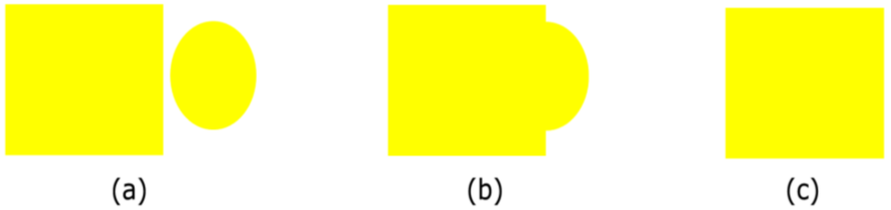


Figure 2.2 The process of camouflage.

Clutter is commonly cited as a key problem in visual tracking. Clutter occurs when multiple objects surrounding the target share similar features with the target. This can be thought of as multiple, partial camouflaging of the target, and causes confusion as to the location of the true target.

Camouflage and occlusion can cause the tracker to become dissociated from its target, so that the data it produces is unrelated to the tracker's behaviour. Therefore, effective handling of these events is vital to the success of a tracking process.

2.2.2. A review of occlusion and camouflage handling

Occlusion and camouflage are inevitable in visual tracking, and a variety of approaches have been adopted in response. Literature search has shown that occlusion can be handled in one of three ways: *improving representation*, *improving search* or through *occlusion reasoning*.

When *improving representation*, one approach is to finesse such problems by careful camera placement (Behera et al., 2012), though this is often not possible (Mittal et al., 2003). Other approaches attempt to improve the observation model used by incorporating additional, learned information (e.g. information of the target(s) visibility and existence within the scene) into its representation (Pérez et al., 2005) or updating the observation model whenever occlusion occurs (Hanzi et al., 2007). Some approaches use a template update (Nguyen et al., 2004) or even use temporal relationships between frames (Verma et al., 2003) to handle occlusion. (Babenko et al., 2009) handled the occurrence of occlusion in face detecting by updating the appearance model using an online multiple instance learning approach. In (Jepson et al., 2003, Jepson et al., 2001),(Han et al., 2005) and (Ross et al., 2004),occlusion is handled via an online adaptive appearance model. While each of these seeks to handle occlusion they do not completely solve the problem. Learned appearance models can improve the accuracy of normal tracking, but require sound data from which to learn. Even if the tracker maintains its link to the target, both full and partial occlusion typically reduce the accuracy of positional estimates. This reduces the quality of the appearance data passed to any learning or update algorithm, and is highly likely to lead to an inappropriate appearance model being learned.

A more direct approach to handling occlusion involves choosing more complex representations of the target object(s) which are comparatively robust to occlusion, as done in (Jeyakar et al., 2008). In (Collins et al., 2005), occlusion is handled by representing target appearance using histograms of colour filter bank responses applied to red, green and blue pixel values within local image windows. While in (Kwak et al., 2011), occlusion is explicitly detected by dividing the target into several cells and training a classifier using a patch likelihood. Some like (Nummiaro et al., 2003, Bullock et al., 2004) try overcoming occlusion via the integration of colour distributions or the use of colour and motion cues within the tracking framework. Increasing the descriptive power of the appearance model increases the likelihood that the tracker will survive partial occlusion but, whatever pattern of features is used, there remains a chance that key information will be hidden. More descriptive appearance models also make it easier to re-acquire the target after full occlusion, but this is only possible if the tracker models target motion well enough to be able to predict where it will reappear.

Meanwhile, the use of mixed dynamical motion models (Brasnett et al., 2007) has also shown to be robust when dealing with occlusion. This, however, increases the complexity of the models as stronger assumptions about the tracked object are imposed to achieve better performance, generally restricting the applicability of the tracker concerned (Senior et al., 2006). The approach is also still unlikely to solve the occlusion problem completely as partial and full occlusion reduces the accuracy of positional estimates, resulting in lower quality motion models being produced. However, it can lead to a system which degrades gracefully in the event of occlusion.

Another way of handling occlusion is by modifying the underlying search engine (Ma et al., 2009). (Lanz, 2006, Arnaud et al., 2007) handles occlusion by modifying the diffusion step within the probabilistic propagation process of a Sequential Monte Carlo method using partial linear Gaussian models. (Song et al., 2010) handles occlusion via a set of rules of tracklet estimation which is embedded into a stochastic graph evolution framework. In (Karavasilis et al., 2011), occlusion is handled by forwarding a prediction of the object's location to a Kalman filter whose parameters are estimated online based on a recent history of the motion models. Others try to handle occlusion by using multiple trackers' output (Leichter et al., 2006, Kwon et al., 2011) or multiple tracking hypotheses (Maggio et al., 2009, Babu et al., 2007, Babu et al., 2006) to help improve the propagation process between time steps. Though this may be effective nevertheless, it will result in a constraint on computational time and resources.

Occlusion reasoning meanwhile, can be implemented by using spatio-temporal reasoning to determine the consistency of dynamic scene interpretation (Bennett et al., 2008), via Bayesian networking (Town, 2007) or by deriving a likelihood model according to image formation principles and implementing occlusion reasoning at pixel level (Lanz, 2006, Arnaud et al., 2007). In (Lascio et al., 2013), the authors uses contextual reasoning to deal with complex occlusion involving a plurality of moving people simultaneously, where the rationale is grounded on a suitable representation and exploitation of the recent history of each moving person being tracked. While (Adam et al., 2006) and (Chockalingam et al., 2009) handles occlusion by using robust statistics to reason about occlusion via decomposing the target into multiple components or patches. Conventionally, occlusion reasoning is done over depth or trajectories. In occlusion reasoning over depth, several projects have sought to address the problems caused by occlusion by explicitly representing occlusion relationship between objects. The goal here is to identify when a tracked target is likely to be (partially) occluded and vary the tracker's operation accordingly. Occlusion reasoning typically involves some estimation of the depths of targets and potentially occluding objects. (Greenhill et al., 2008) recover a probability density function for scene depth at each pixel from a training set of observations of people moving through an indoor environment. This is then used in a correlation-based tracker to prevent occluded pixels being included in the computation of the correlation measure. The approach successfully reduces the effect of occlusion by static, background objects. To deal with the dynamic occlusion caused by other moving objects, (Greenhill et al., 2008) adopt the approach of (Senior et al., 2006). This exploits appearance models of the objects being tracked to allow each pixel to be assigned to the most likely object, and so relative depth to be recovered, when two targets overlap. When occlusion reasoning over trajectories, (Rosales et al., 1998) detected dynamic occlusion by projecting the trajectories estimated by an Extended Kalman filter forward in time and thresholding the expected degree of overlap between pairs of targets. Occlusion is confirmed by applying a similar test to actual target descriptions. Targets are then merged for the duration of the occlusion, which can also be predicted using the available information, and split again once it’s over.

Explicit discussion of handling camouflage has received less attention in the vision literature as compared to occlusion. In (KaewTrakulPong et al., 2003), they describe a system which employs colour, motion and shape models to track low resolution targets. The shape model used is a simple bounding box, which they note increases in size during camouflage and shrinks during occlusion. When these effects occur, the tracker moves from a data association to stochastic sampling process. It should be stressed, however, that (KaewTrakulPong et al., 2003) do not explicitly seek to detect these events and the switch to stochastic sampling is made whenever a new observation cannot be associated with an existing track, whatever the reason. Whereas, in (Stolkin et al., 2012, Talha et al., 2012, Zhou et al., 2012, Shen et al., 2012), camouflage is handled by consistently improving the target state estimation at each successive frame through comparison between the foreground and background models. (Stolkin et al., 2012, Talha et al., 2012) presented a method that combines image data from a colour camera and a deep infra-red thermal imaging camera which continuously relearns local background models in each imaging modality, comparing these against a model of the foreground object being tracked and thereby adaptively weighting the data fusion process in favour of which ever imaging modality is the most discriminating at each successive frame. In the event of camouflage, the method reduces the influence of this poor modality and relies more on information from more discriminating modality. (Zhou et al., 2012), presents a novel foreground object detection scheme which constructs a foreground model based on the object model and the state of each target using the EM framework.

Camouflage is handled by augmenting the foreground detection using the foreground model, whereby, the fusion of the detection result for estimating the object's state avoids the estimation drifting to the background area. (Shen et al., 2012) proposed a better temporal constraint to encourage segmentation which maintains a consistent appearance of foreground in consecutive frames, as this assists shape prior to alleviate camouflage. It should be stressed again here that, even though, these methods do attempt to handle camouflage when and if it does occur, these methods were not created to explicitly seek and detect the occurrence of camouflage during tracking.

Examination of the literature has shown no significant analysis of the changes taking place within a tracker during occlusion and camouflage, i.e. over the few frames it takes for the target to transition from normal visibility to full occlusion or camouflage. Closer examination of the measurement stage of a particle filter-based tracker provides useful information on the changes taking place within the particle set during occlusion and camouflage events. Measurement steps are often quite complex, powerful operations but can provide a lot of information about the target's local environment. In addition, particle filters are particularly good at providing this information, as they sample from both the target and its surroundings.

Section 2.3 presents a review of two common model estimation concepts used to build a tracking engine. The section will conclude by justifying the selection of a particle filter-based tracker as the platform for the research reported in this thesis.

2.3. A review of model estimation concepts

Two model estimation concepts commonly used to build a tracking engine, the Kalman filter and particle filter, are examined in this section.

2.3.1. Kalman filter

The Kalman filter was introduced in 1960 by (Kalman, 1960), but its roots can be traced as far back to the Gauss's method of least squares in 1795 (Simon, 2001). Since its discovery, the Kalman filter has been applied in a range of diverse applications. In spite of it being developed for space navigation, the Kalman filter has been used in areas such as nuclear power plant instrumentation, demographic modelling, manufacturing, the detection of underground radioactivity, fuzzy logic, neural networks (Simon, 2001), automated missile guidance systems, robotics (Nagenborn, 2003), radar tracking, sonar ranging, satellite orbit determination (Cipra, 1999) and improving precision (Greenspan et al., 2004). The Kalman filter is also used to build tracking engines within visual trackers (Forsyth et al., 2003, Welch et al., 1995).

Literature has shown the Kalman filter being applicable to a diverse of problems, though, it is limited to linear motion and uni-modal Gaussian densities. The Kalman filter cannot represent multiple alternative hypotheses. Implementing a Kalman filter is also made more difficult due to the existence of the Riccati equation (Isard et al., 1998a). Moreover, Kalman filter based contour trackers which run in real time are very susceptible to distraction by clutter and correlation-based systems and is vulnerable to changes in object appearance and lighting and rapidly slows down as the space of deformations increases in complexity (Isard et al., 1998b). As a result, an alternative solution to overcome the limitations of Kalman filter is provided by a group of algorithms based upon the notion of the particle filter.

2.3.2. Particle filters

Particle filters, also known as sequential Monte Carlo methods (Kitagawa, 1996, Doucet et al., 2000, Cappe et al., 2007), are powerful estimation techniques based on simulation commonly used to handle non-Gaussian densities. Particle filters are based on point mass or "particle" representations of probability densities which is applicable to any state-space method and generalize the traditional Kalman filtering method (Arulampalam et al., 2002). The Particle filter is defined as a class of simulation filters that recursively approximate the random variable $a_t \mid Y_t = (y_1, \ldots, y_t)'$by "particles" $a_t^1, \ldots, a_t^M$, with discrete probability mass of $\pi_t^1, \ldots, \pi_t^M$ (Pitt et al., 1999). The principle advantage of particle filters is that they do not rely on any local linearization technique or any crude function approximation and they have found real time application in fields as diverse as chemical engineering, computer vision, financial econometrics, robotics (Doucet et al., 2008), statistics, and signal processing (Cappe et al., 2007). Particle filters can be subdivided into two categories: particle filters that re-use particles and require re-sampling to prevent divergence, e.g. sequential importance particle filters and Bootstrap particle filters, or particle filters that do not re-use particles and therefore require no re-sampling, e.g. Gauss-Hermite particle filters, Monte Carlo particle filters and Unscented particle filters (Haug, 2005). Particle filters are simple to implement, robust towards fluctuating motions and able to handle non-linearity. As a result, a variety of algorithms utilizing the concept of particle filter together with some other variation of concepts has been introduced. Among them are Gaussian particle filters (Kotecha et al., 2003), the Rao-Blackwellised particle filter (Sim et al., 2007), the Kalman particle filter (Li et al., 2003), Auxiliary particle filter (Pitt et al., 1999) and the mean shift embedded particle filter (Shan et al., 2007).

In real tracking, multi-modal distributions are often required to represent competing hypotheses. Particle filtering is used to model this situation as particle filters can represent a higher degree of ambiguity in the target's state. Particle filtering was introduced into computer vision in the Condensation algorithm (Isard et al., 1996, Isard et al., 1998a).

2.3.2.1. The Condensation algorithm

Condensation was developed to handle non-Gaussian state densities in visual tracking. Condensation is part of the particle filter family, though there is a significant difference that separates the two. (Li et al., 2003) presents evidence that for the Condensation sampling step, the proposal distribution from which particles are drawn is the distribution conditional on the particle state at the previous frame: the latest observation is only used in the weighting step and not in the sampling step.

Each particle in Condensation represents a single hypothesis as to the target's state, with some probability of being correct. Particles are compared to an observation in order to predict where the target is likely to be in the following frame of an image sequence. In visual tracking, particles are normally shown as spots of different colour or intensity overlaying a series of images as shown in Figure 2.3.

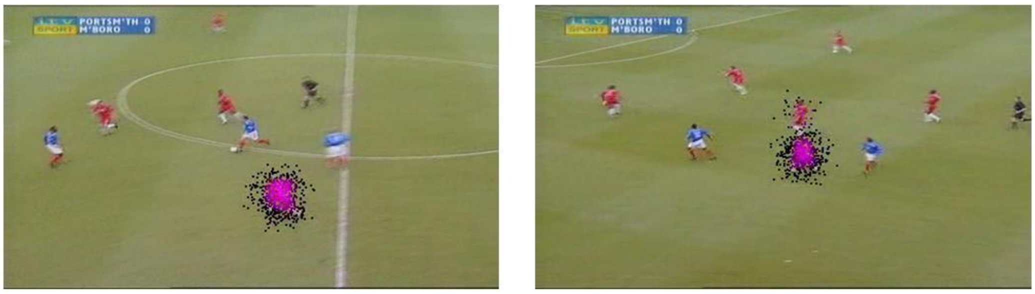


Figure 2.3 Particles tracking a football player.

Condensation is based on factored sampling. Factored sampling is used to approximate probability distribution, whereby in principle can be evaluated using the Bayes' rule (Bayes et al., 1763). Though, in practice, it can be evaluated using iterative sampling techniques (Isard et al., 1998a). Condensation is estimated by a set of discrete particles. A sample or particle set x_t^n which consists of the target's state vector representation is chosen from a prior distribution, denoted as $\{(x_t), n = 1, \ldots, N\}$, where *n* refers to the *n*th particle set, *t* refers to the current time step and *N* is the sample size of the population. Target state is commonly position but need not be as shape, colour or velocity can be used as alternative. Particles drift between time steps, so a motion model is used. In addition, to capture the uncertainty in the motion model, some noise is added to the particles. This noise is added to the target's state vector representation. The particles then diffuse and agitate individually. Particles are distributed across the search space as each particle experiences Brownian motion step independently. This results in a new un-weighted particle set for the new time step. Subsequently, factored sampling takes place whereby the weights π_t for each particle are obtained based on how similar the particle is when matched against an observation density. Consequently, this result in a weighted particle set, denoted as $\{(x_t^n, \pi_t^n), n = 1, \ldots, N\}$. A high similarity will result in a high ratio, whereas, a low similarity will result in a low ratio value being assigned. To start the next iteration, a posterior distribution set is estimated by copying particles from the prior distribution set but taking into account their respective weights. Therefore, particles with high weight can be selected more than once. The sample size for the posterior distribution is identical to the prior distribution for time *t+1*. Finally the new posterior distribution set replaces the old prior distribution set and the entire process is repeated again for the next time step.

The advantages presented by Condensation have resulted in several extensions to Condensation being addressed in the literature. ICondensation (Isard et al., 1998b) provides a combination of low and high level information in a constraint probabilistic framework with the incorporation of importance sampling into Condensation. Mixed-state Condensation (Isard et al., 1998c) meanwhile, develops random sampling methods to allow automatic switching between multiple motion models as a natural extension of the tracking process.

Section 2.4 provides groundwork for the research reported in this thesis. The section will begin by detailing the implementation of the Condensation algorithm. Subsequently, test image sequences that exhibit occlusion and camouflage are presented. The section will conclude by analysing the results gathered from applying the Condensation algorithm to test image sequences exhibiting occlusion or camouflage.

2.4. Groundwork: Condensation, occlusion and camouflage

2.4.1. Condensation algorithm implementation

A flowchart of the Condensation process is shown in Figure 2.4, further details are given in Algorithm 2.1. Implementation of Condensation tracking is straightforward. The particle set is represented as an array of N simple data structures, each representing one particle. Each particle data structure contains a hypothesised target state and an associated weight. State values are initialised randomly, and initial weights computed by matching the hypothesised states (typically target position) to the first image in the sequence. To predict target state (location) in the next image, N particles are randomly selected from the array. The selection process takes weights into account, so higher weighted particles are more likely to be selected. A given particle may be selected more than once. The selected particles are projected forward in time using a model of the expected motion of the target, often constant velocity. Random noise is then added to the predicted state descriptions. This prevents multiple copies of the same particle from making identical hypotheses and increases the search area. The new particle are weighted, as before, by comparing hypothesised states to image data; the next image in the sequence. The set of weighted particles created at each time step (i.e. for each image) represents the tracker's estimation of it's targets properties. For display purposes, the highest weighted particle or a weighted mean of the particle set can be used to indicate target location. Examples of the operation of the Condensation algorithm are shown in section 2.4.3.

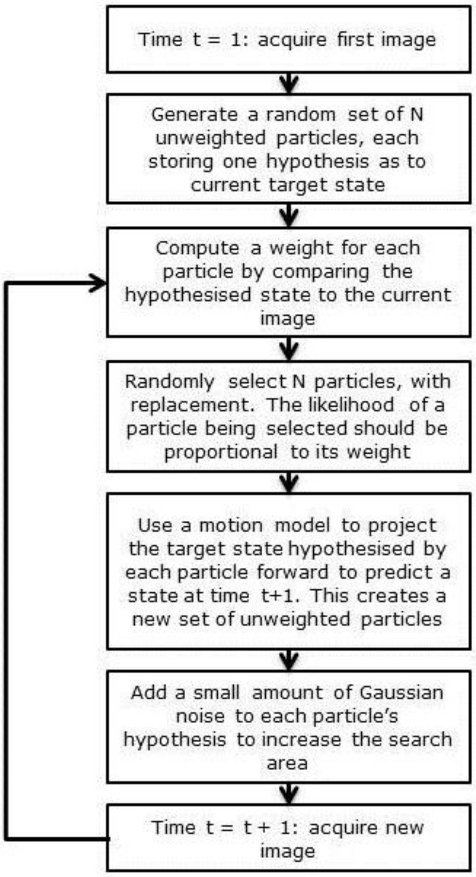

Time t = 1: acquire first image
Generate a random set of N unweighted particles, each storing one hypothesis as to current target state
Compute a weight for each particle by comparing the hypothesised state to the current image
Randomly select N particles, with replacement. The likelihood of a particle being selected should be proportional to its weight
Use a motion model to project the target state hypothesised by each particle forward to predict a state at time t+1. This creates a new set of unweighted particles
Add a small amount of Gaussian noise to each particle's hypothesis to increase the search area
Time t = t + 1: acquire new image

Figure 2.4 A flowchart of a Condensation process.

CONDENSATION ALGORITHM

1) **Randomly select** N samples $\{X^{(n)}\}_{n=1}^{N}$ of particles from the previous time-step sample set $\{X_{t-1}^{(n)}, \Pi_{t-1}^{(n)}\}_{n=1}^{N}$, with probability relative to their weight, $\Pi_{t-1}^{(n)}$

2) **Propagate** forward the sample set by sampling from:

$$P(X_t \mid X_{t-1})$$

3) **Add noise** using a probability distribution function:

$$pdf = \frac{1}{\sqrt{2\pi\sigma^2}} e - \frac{(x-\mu)^2}{2\sigma^2}$$

4) **Measure** and **weight** each particle based on a measurement using a colour observation model:

$$\Pi_t^{(n)} = P(Z_t \mid X_t^{(n)})$$

Algorithm 2.1 Condensation algorithm (Isard et al., 1996; Isard et al., 1998a).

A point representation (Yilmaz et al., 2006) Condensation algorithm is implemented in this thesis. A sample state is defined as:

$$X = (x, y, u, v), \Pi$$

where x and y are the target coordinate location, u and v are the target velocities and Π is the sample weight.

The success of the Condensation algorithm is governed by the effectiveness of the measurement model, motion model and initial state values. The following section details the measurement and motion model(s) and justification of initial state values.

2.4.1.1. Measurement model

The measurement model estimates the probability of a target being present at the location being examined. The probability of a given state is estimated by matching the measurement model to the current model of the image at a particular location. An observational measurement model (measurement model henceforth) represents the original state of the target. Feature selection on which the measurement model and current model is built on plays a pivotal role in the success of determining the presence of the target in the image during tracking. Different feature representation such as colour, edges, optical flow or texture (Yilmaz et al., 2006) can be used to characterize the target. Nevertheless, colour representations based on RGB colour space or HSV colour space have shown (Nummiaro et al., 2003) to be a common approach for building these model(s). HSV colour space is more illumination invariant than RGB colour space. By discarding the V value, hue and saturation becomes less sensitive to illumination changes. As mentioned in Chapter 1, illumination changes in tracking can gravely affect the tracking process. Therefore, the proper choice of colour space is vital to the success of tracking. As a result, the HSV colour space is used in this thesis. The measurement model and current model are built as a two dimensional histogram. The histogram is a distribution representation of the target's hue and saturation. Hue and saturation can be computed using (2.1) and (2.2) respectively. The colours red, green and blue are represented as R, G and B, while hue and saturation are represented as H and S respectively in (2.1) and (2.2).

$$H = \begin{cases} \left(0 + \frac{(G-B)*60}{(MAX(R, MAX(G,B)) - MIN(R, MIN(G,B)))}\right), if \ R = MAX \\ \left(120 + \frac{(B-R)*60}{(MAX(R, MAX(G,B)) - MIN(R, MIN(G,B)))}\right), if \ G = MAX \\ \left(240 + \frac{(R-G)*60}{(MAX(R, MAX(G,B)) - MIN(R, MIN(G,B)))}\right), if \ B = MAX \\ 0, if \ MAX = 0 \end{cases} \tag{2.1}$$

$$S = \frac{(MAX(R, MAX(G,B)) - MIN(R, MIN(G,B)))}{MAX(R, MAX(G,B))} \tag{2.2}$$

The similarity between the measurement histogram and current histogram is measured using Bhattacharyya distance (Bhattacharyya, 1943, Kailath, 1967, Djouadi et al., 1990, Ahere et al., 1997, Comaniciu et al., 2003). Bhattacharyya distance measures the similarity of two normalized discrete or continuous probability distributions. In this thesis, $p^{(u)}$represents the measurement histogram while the current histogram is represented by $q^{(u)}$. The result obtained from measuring the similarity of $p^{(u)}$and $q^{(u)}$ is a value between 0 and 1. An exact similarity between $p^{(u)}$and $q^{(u)}$will produce a result of 1 whereas non similarity will produce a result of 0. The Bhattacharyya distance is computed using (2.3):

$$Bhattachar yya \ distance = \sqrt{\sum_{u=1}^{M} \sqrt{p^{(u)} q^{(u)}}} \tag{2.3}$$

where $p = \left\{p^{(u)}\right\}_{u=1,\dots,M}$ and $q = \left\{q^{(u)}\right\}_{u=1,\dots,M}$. M refers to the total number of histogram bins used.

2.4.1.2. Motion model

Motion models are applied to the target's original position (x, y) to allow the target to experience motion across time. A constant velocity model is commonly used. At each time step, the target's new position (x_t, y_t) is estimated by adding the velocity (u_t, v_t) onto the old target's position (x_{t-1}, y_{t-1}). The velocity at every time step is computed using (2.4):

$$Velocity(u_t, v_t) = Target(x_t, y_t) - Target(x_{t-1}, y_{t-1}) \tag{2.4}$$

Process noise is also added to the target's position to capture the uncertainty in the motion model. In this thesis, noise is generated randomly using a probability density function. The Box-Muller transform (Box et al., 1958) is used to generate a pair of random numbers from the same normal distribution given a pair of random numbers. Since only one noise value can be added to the target's position at a given time, therefore, only one random number is used. The random number is generated using (2.5):

$$Noise = \left(b * \sqrt{\frac{-2\ln c}{c}} \right) \sigma + \mu \qquad (2.5)$$

where b is a given random number, c is the total multiplication of 2 given random numbers, mean, μ is zero and standard deviation, σ is 5. A standard deviation of 5 is selected because 95% of noise will fall between ± 5. Hence, an additional noise of 5 pixels is added to the target's new position after adding the original target position with the velocity model. Any higher standard deviation will cause the target to experience a drastic increase in motion which can result in the failure of tracking.

2.4.1.3. Initial state values

In Condensation, increasing the number of particles typically results in increased tracking performance. However, it also leads to an increased in processing time. As a compromise, a particle set size of 100 per target was used. A particle set size of 100 was found to produce a good tracking result for artificial videos as well for challenging real-world videos.

For each video sequence, the initial starting point is user defined. Initial particles are automatically spread within the target's vicinity using a normal distribution random number generator. A normal distribution is used because 95% of the particles generated will be within one sigma radius of the initial position. This ensures that the particles generated remains within the vicinity of the target boundary and avoids particles being initialized to track something other than the actual target. The normal distribution random number generator details have already been presented in the Motion model section. A standard deviation of 5 is used. Therefore, 95% of particles will be placed approximately ± 5pixels from the user defined position.

Process noise parameters were chosen to distribute particles over an area of radius ranging from 1 to 10 and it is user defined. The selection of a radius size depends on the target size. A smaller radius value is chosen to track a small size target. This is because it will force the particles to stay concentrated onto the target. If a higher radius value is chosen, it may cause some particles to not track the target but instead track the background surrounding the target.

Histogram bins are user defined as well. The number of histogram bins ranges from 10 to 50. A histogram bin of size 10 was found to produce a good tracking result when used on artificial and real videos.

The initial velocities of particles are set to zero.

2.4.2. Test videos

Test video selected or created for this thesis, all exhibit occlusion and/or camouflage. Test videos are divided into two groups: artificial and real test videos.

2.4.2.1. Artificial videos

Artificial videos are created by combining together a series of images where, in each image, a target is placed at different position on a path to exhibit a constant motion. Images were created to show a target (yellow circle) either experiencing: (1) normal tracking or (2) occlusion or (3) camouflage. In all the images created, the image background is always initialised to white.

For normal tracking, a yellow circle is next drawn at the left boundary of the image. Then, in successive images, a constant velocity model is added to the circle's position. The velocity model is specified by the user. Adding the velocity model causes the circle to be drawn at a different position in the image, which simulates the circle moving horizontally. This process is repeated until the circle reaches the right boundary of the image. Finally, all the images are concatenated together to produce the artificial video.

For occlusion and camouflage, the process is repeated. Additionally, in these images, a differently coloured rectangle (to invoke occlusion) or an identically coloured rectangle (to invoke camouflage) is drawn at the middle of every image and on the path of the moving yellow circle. As before, the images are then appended together to produce the respective artificial videos once the circle reaches the right boundary of the image.

A total of six artificial videos were created. Each exhibits one of three different scenarios: a target moving normally, a target experiencing occlusion while moving and a target experiencing camouflage while moving. In order to demonstrate the reliability and effectiveness of the tracker used, the six videos are separated into 2 groups. One group of videos has no clutter present while the second group of videos has 4 types of clutter present. Clutter is added to make the videos more realistic as real videos are never entirely free from clutter.

Artificial videos that have no clutter present show a yellow circle placed at the left boundary of the image sequence moving horizontally to the right boundary of the image. In two videos, a differently (to invoke occlusion) or an identically (to invoke camouflage) coloured rectangle is placed in the path of the moving yellow circle. Figure 2.5 illustrates a target moving normally while Figure 2.6 and Figure 2.7 illustrate a target experiencing occlusion and camouflage respectively.

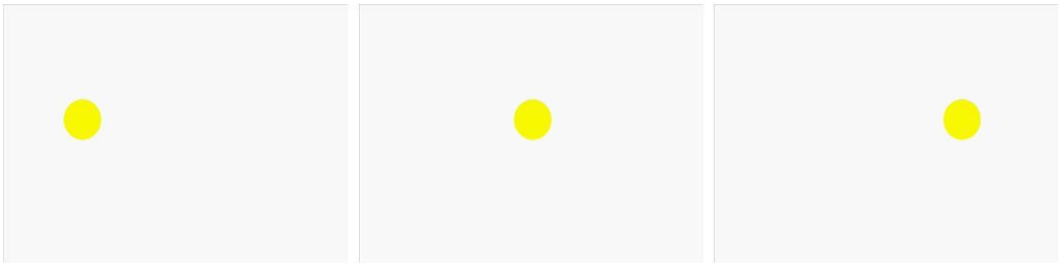


Figure 2.5 A yellow circle moving normally.

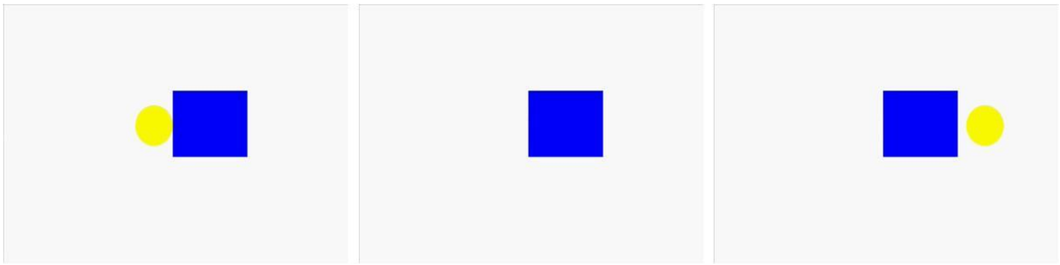


Figure 2.6 A yellow circle experiencing occlusion.

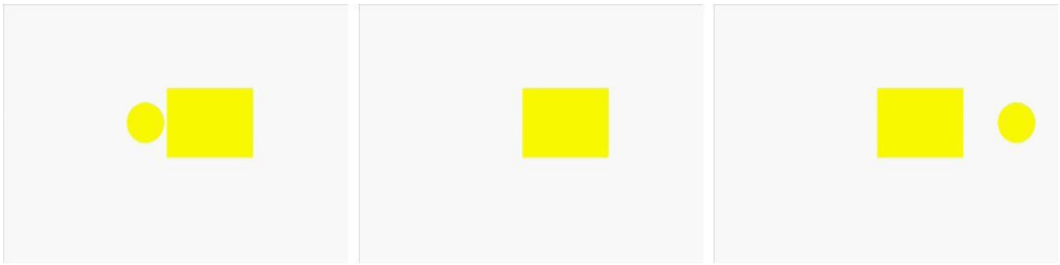


Figure 2.7 A yellow circle experiencing camouflage.

In the artificial videos with clutter, background clutter and three different types of noise are added: motion noise, image noise and target noise. Background clutter adds objects with different sizes, shapes and colours randomly to the background of the image while the target is moving in the foreground of the image. Motion noise adds random velocity-(u, v) to the target's position-(x, y) in order to have an uneven interval in the target's movement. Image noise adds random colour pixels onto the entire image sequence of a video. Target noise only adds random colour pixels to the target's vicinity in the image sequence. Motion, image and target noise are generated using a probability density function with a mean, μ of zero and standard deviation, σ of 10. A standard deviation of 10 is selected so that 95% values obtained will fall between ± 10. Since a percentage value is used to control the amount of noise generated for each artificial video, therefore, if a higher standard deviation value is selected, it will contribute to excessive noise being generated which will affect the tracking process. Trial and error using percentage values ranging from 0.1% to 100% was performed to determine the optimum percentage for each noise. Optimum percentage values were chosen on the basis of optimum values of noise generated does not affect the tracking process at an early stage even before the target experiences occlusion or camouflage. However, if no percentage limit is found, half of the maximum percentage is selected as the optimum value. The maximum percentage is not used because an overly noisy video will be produced since each video generated comprises of a combination of all four noise types. Therefore, the optimum values for each noise are: background clutter (50%), image noise (50%), movement noise (1.0%) and target noise (0.14%). In tracking, the measurement model which determines the success of tracking a target is build during the initial frame of a video sequence. To avoid any form of noise also being recorded into the measurement model which can affect the tracking process acutely, no form of noise is generated in the initial frame of the artificial videos. Figure 2.8 illustrates a yellow circle moving normally but with clutter present while Figure 2.9 and Figure 2.10 demonstrates a yellow circle experiencing occlusion and camouflage in addition to the presence of clutter.

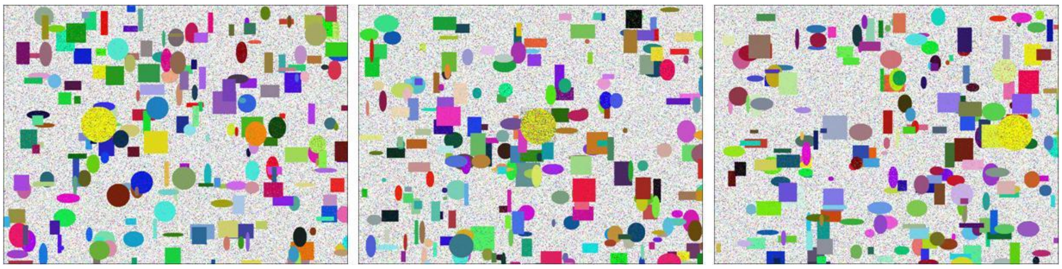


Figure 2.8 A yellow circle moving normally but in the presence of clutter.

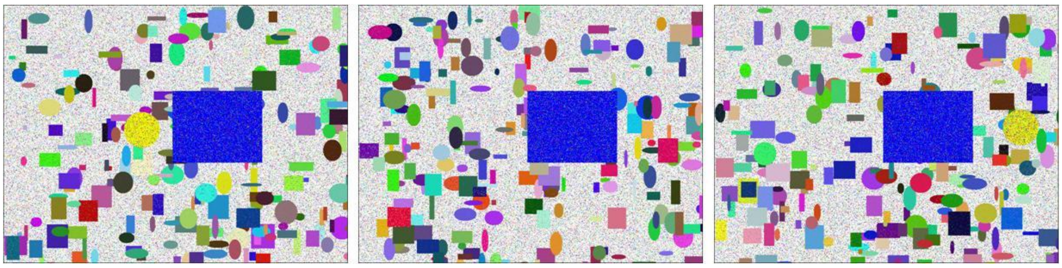


Figure 2.9 A yellow circle experiencing occlusion and in the presence of clutter.

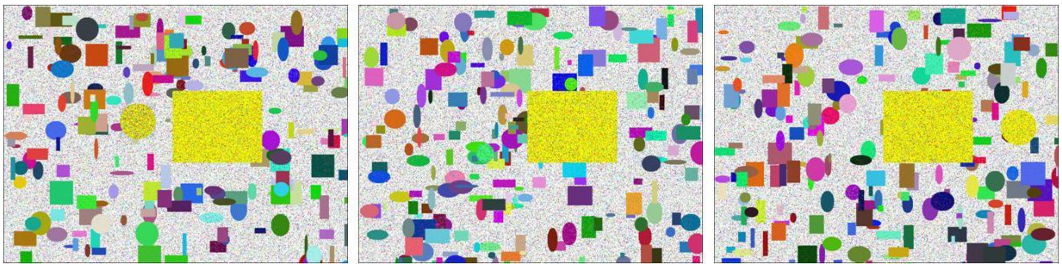


Figure 2.10 A yellow circle experiencing camouflage and in the presence of clutter.

2.4.2.2. Real videos

A total of six real videos was chosen or captured. Similar to the artificial video, each of the real videos exhibits a target either moving normally, or experiencing occlusion or camouflage while moving.

In Figure 2.11, a tree occludes a woman while in Figure 2.12, a player camouflages a football. Figure 2.13, Figure 2.14 and Figure 2.15 demonstrate a tennis ball either moving normally or experiencing occlusion or camouflage. Figure 2.16 illustrates a table tennis bat experiencing self-occlusion.

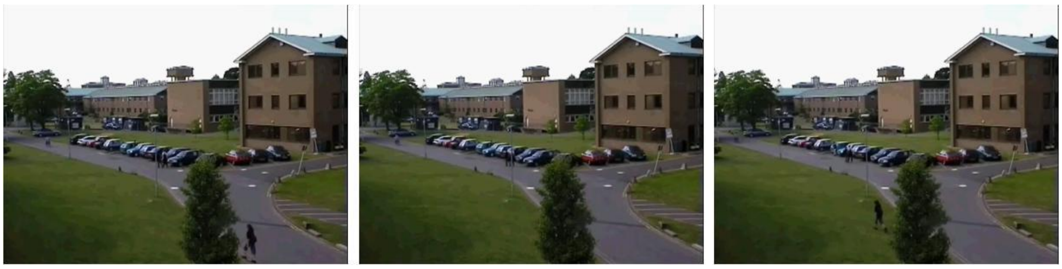


Figure 2.11 A tree occludes a woman.

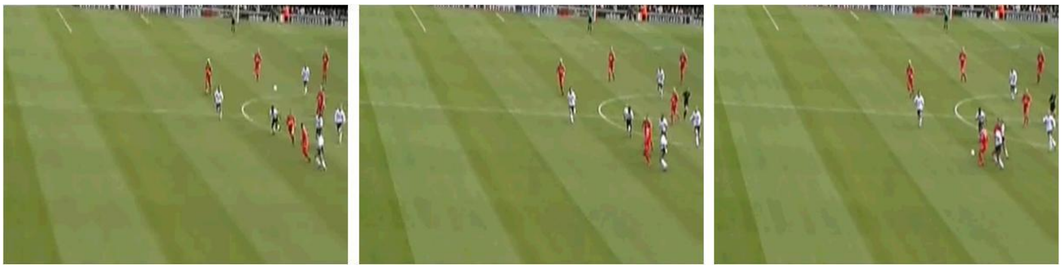


Figure 2.12 A player camouflages a football.

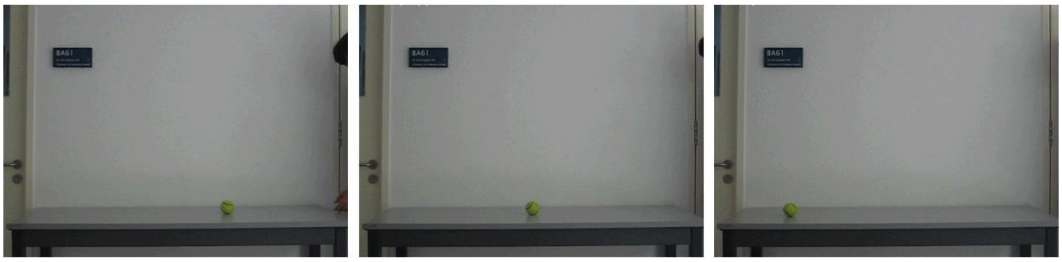


Figure 2.13 A tennis ball moving normally.

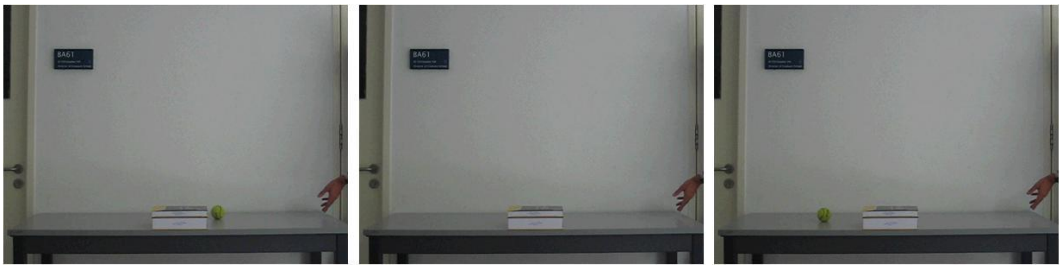


Figure 2.14 Some books occlude the tennis ball.

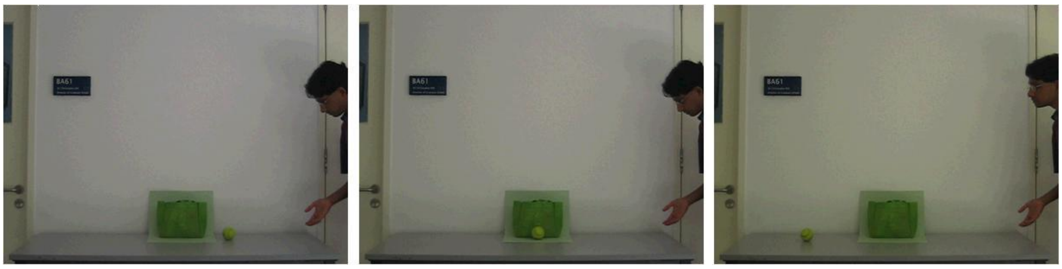


Figure 2.15 A bag and file camouflages the tennis ball.

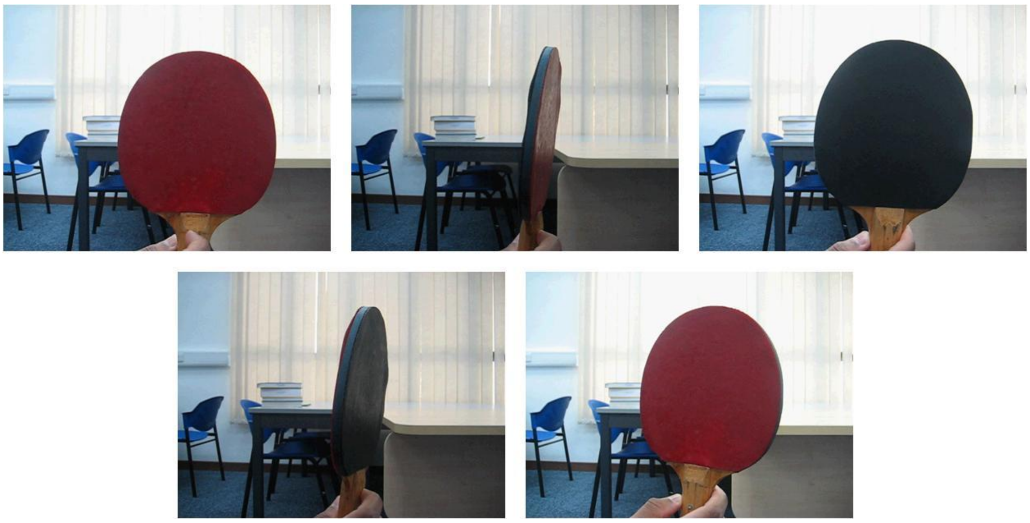


Figure 2.16 A table tennis bat experiences self-occlusion.

2.4.3. Condensation: the effect of occlusion and camouflage

Applying a particle filter or Condensation algorithm to the test videos allows a closer examination of the measurement stage to be made. Analysis is centred on the particle distributions that arise as tracked targets become occluded or camouflaged. It should be pointed out here again that the detection of occlusion and/or camouflage is focused on target object(s) moving in 2D.

2.4.3.1. Tracking parameters

In all the test videos, a particle set size of 100 was used. Process noise varied for each video. Depending on the target size, a radius size ranging from 1 to 10 is selected to produce a good tracking result under normal conditions. A histogram bin size of 10 is selected for all the test videos.

2.4.3.2. Normal tracking

In Figure 2.17 and Figure 2.18, the particles successfully track the yellow circle from the start of the tracking process to the end. The particle set also successfully tracks the tennis ball in Figure 2.19 from the start to the end of the video sequence.

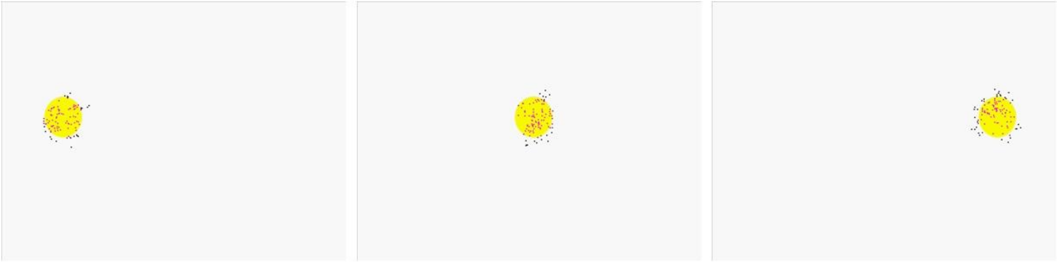


Figure 2.17 Particles successfully track the yellow circle.

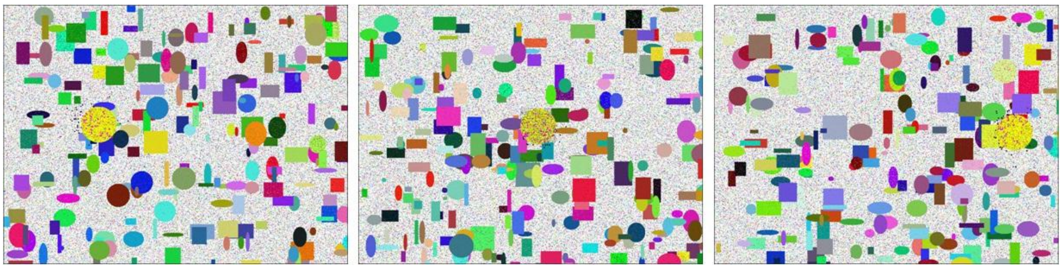


Figure 2.18 Particles successfully track the yellow circle amid clutter.

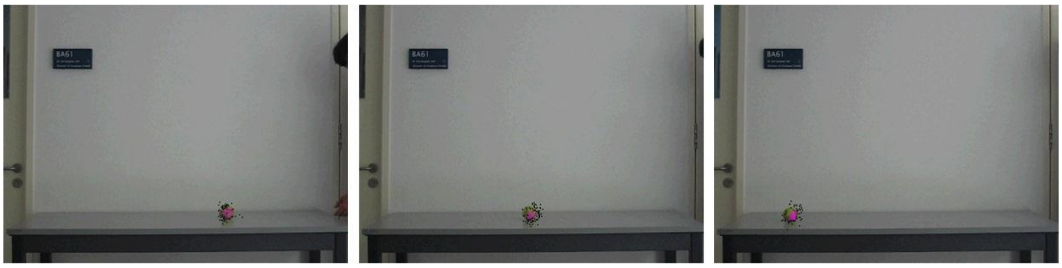


Figure 2.19 Particles successfully track the tennis ball.

2.4.3.3. Occlusion

Analysing Figure 2.20 and Figure 2.21 demonstrates that as the yellow circle becomes occluded by the blue rectangle, some particles get transferred onto the blue rectangle and die, while the remaining particles start to experience some form of clustering. After losing the target, the particles are tightly clustered and move rapidly, but randomly, over the image plane. However, though the yellow circle reappears after being occluded, the particles never reattach themselves onto the target.

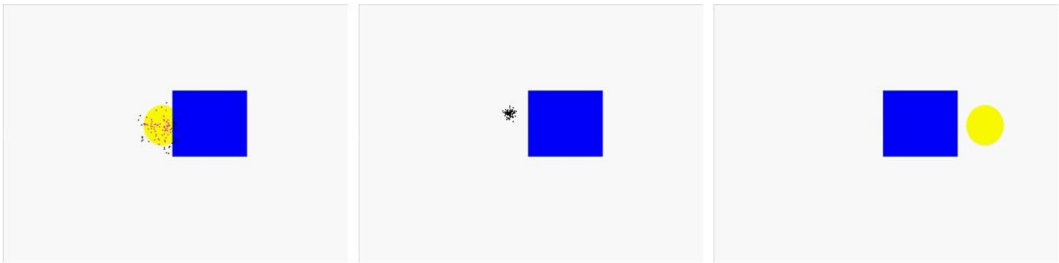


Figure 2.20 Particles lose the yellow circle following occlusion.

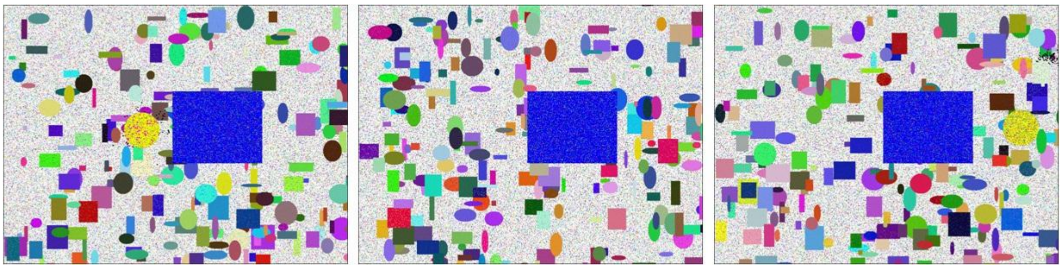


Figure 2.21 Particles stop tracking the yellow circle as occlusion occurs amid clutter.

The particles also behave similarly in real videos as depicted in Figure 2.22, Figure 2.23 and Figure 2.24. In Figure 2.22, as the tree occludes the woman, the particles cluster together and eventually stop tracking the woman when she is totally occluded by the tree. The particles fail to reattach themselves after the woman emerges from behind the tree. When the tennis ball is occluded by some books in Figure 2.23, the particles once again become closely clustered together. They again fail to reattach themselves onto the tennis ball when it reappears. In Figure 2.24, the table tennis bat experiences self-occlusion. As the table tennis bat rotates, the particles tracking the red side of the bat get tightly clustered together as the red side of the bat slowly disappears from view. When the target is fully self-occluded, the particle set is seen closely clustered together and no longer tracking the bat. The particles fail to reattach themselves to the red side of the table tennis bat once it returns to view.

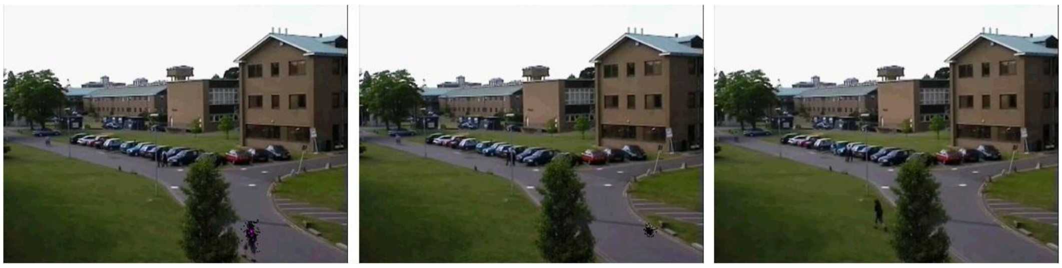


Figure 2.22 Particles stop tracking the woman as she becomes occluded by the tree.

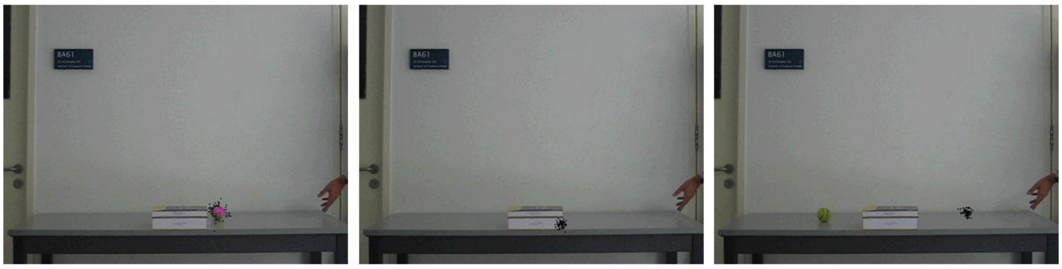


Figure 2.23 Particles stop tracking the tennis ball as some books occlude the tennis ball.

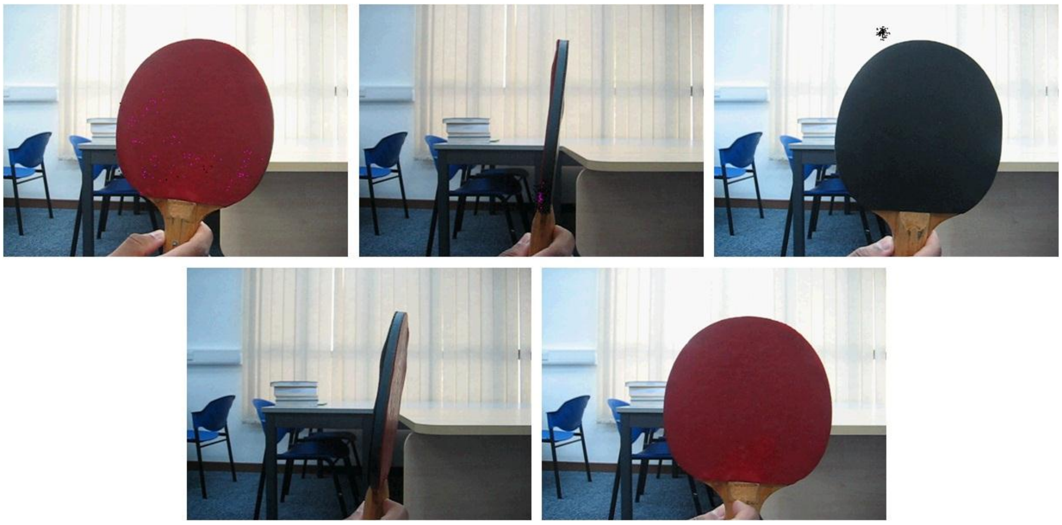


Figure 2.24 Particles stop tracking the table tennis bat as the table tennis bat experiences self-occlusion.

2.4.3.4. Camouflage

Examining the particles' behaviour in Figure 2.25 and Figure 2.26, show that as the yellow circle becomes camouflaged by the yellow rectangle; particles are transferred onto the rectangle and start spreading out. The particles never reattach themselves to the circle but instead remain fixed to the camouflaging object.

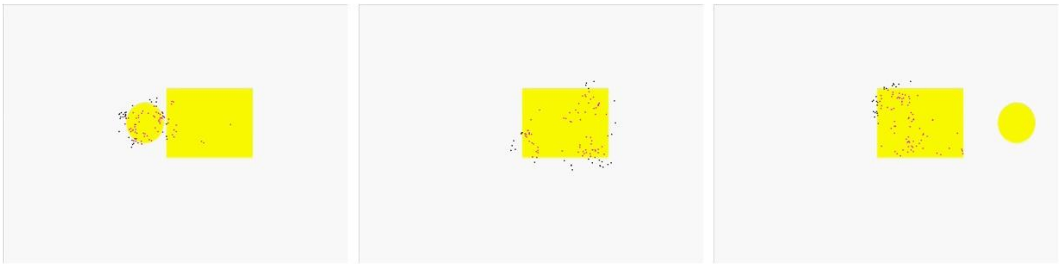


Figure 2.25 Particles get transferred and start tracking the yellow rectangle as the yellow circle experiences camouflage.

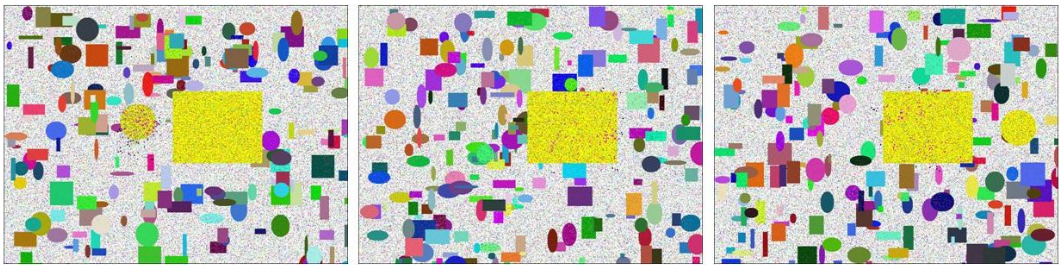


Figure 2.26 Particles never reattach themselves onto the yellow circle again after it experiences camouflage.

Figure 2.27 and Figure 2.28 depict the particles' behaviour in real videos. In Figure 2.27, as the player camouflages the football, particles tracking the football get transferred onto the player. Particles never again track the football, even after it reappears. The same behaviour is also illustrated in Figure 2.28 as the tennis ball is camouflaged by a bag and file. The particle set gets transferred onto the camouflaging bag and file and begins to spread widely within them. Though the tennis ball reappears from camouflage, the particles never reattach themselves to it.

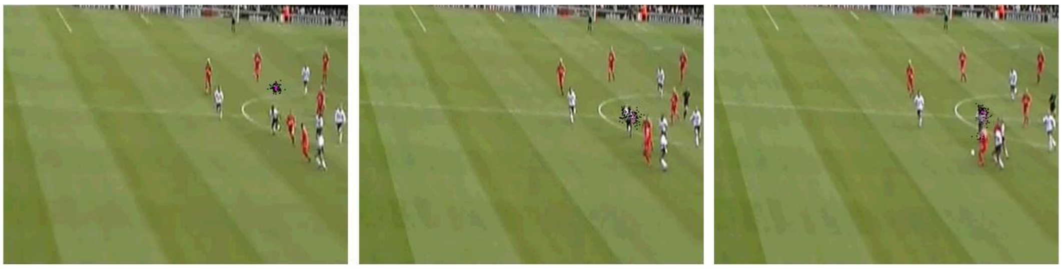


Figure 2.27 Particles get transferred onto the player and field line as the football becomes camouflaged by the player.

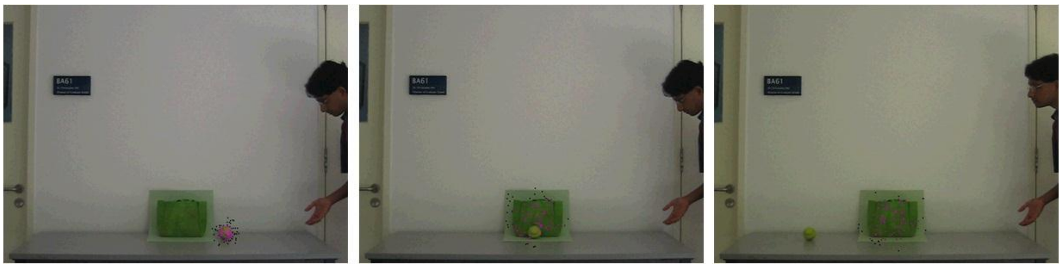


Figure 2.28 Particles never again reattach themselves onto the tennis ball even after the tennis ball stop experiencing camouflage. Particles continue to track the camouflaging bag and file instead.

2.4.3.5. Summary

Analysis of all test videos leads to the following hypotheses:

- When occlusion occurs, the particles become closely clustered together and stop tracking the target. The particles typically fail to reattach themselves to the target when it reappears, though they may of course do so by chance.
- When camouflage occurs, the particles are transferred onto and start to spread widely within the camouflaging object. Again, the particles never reattach themselves to the target.

2.5. Chapter summary

Kalman filters or Sequential Monte Carlo methods, e.g. particle filters are model estimation concepts commonly used to build a tracking engine. Though, literature has shown that the Kalman filter is proven to converge, the method is limited to linear motion and uni-modal priors. As a result, particle filtering is used as an alternative solution. Particle filters can represent multi-modal distributions by way of multiple hypotheses which represent some features (e.g. colour, texture or contour, shape) of the target in question. Customarily, these hypotheses are represented as a set of discrete particle that are often shown as spots of different colour or intensity overlaying the target on the image data. While many effective tracking algorithms have been based on the particle filter, occlusion and camouflage remain problems. Literature has detailed that previous tracking techniques have sought to avoid these problems by keeping the tracker more tightly focused on the target, so that camouflage is simply not noticed and the tracker is more likely to reacquire the target following (short periods of) occlusion. Multiple motion models have also been used, as have more detailed texture and colour cues that better model the target appearance. Alternatively, areas of the environment in which the target is more likely to appear might be identified. While these approaches improve tracker performance, nevertheless, they do not entirely remove the problems of occlusion and camouflage. This thesis takes an alternative approach, arguing that the solution lies not in avoiding, but in detecting and explicitly reacting to these disruptive events. A particle filtered-based algorithm, Condensation is applied onto test videos that explicitly exhibit occlusion and/or camouflage. Analysis results provide evidence that particle set can be analysed collectively to provide vital information about the target's state during the occurrences of occlusion and camouflage.

These new findings are explored further in Chapter 3 and Chapter 4 of this thesis.

Chapter 3 A Particle Clustering Approach

3.1. Motivation

In Chapter 2, two key observations are drawn from an analysis of the effects of occlusion and camouflage on the operation of the Condensation algorithm:

1. During occlusion, the particle set is compressed. After the reappearance of the target the closely clustered particles typically fail to reattach themselves to the target, though they may of course do so by chance.
2. During camouflage, particles are transferred onto the camouflaging object and the particle set begins to spread across the camouflaging object. As the target reappears, the widely spread but inappropriately placed particles typically fail to reattach themselves to the target.

The hypothesis considered here is that close examination of the behaviour of particle distributions every time step can provide an indication of the occurrence of occlusion and camouflage. A key feature of particle sets is that they are multi-modal: a given particle set will dynamically arrange itself into a (usually small) number of clusters, each representing a different, conflicting interpretation of the available data. The number, size and spread of these clusters varies with the relationship between target and its surroundings. When attempting to quantify the effect occlusion and camouflage have on a particle set, it is natural to apply a clustering algorithm and use the parameters of the resulting clusters to characterise the particle distribution.

3.2. Aim

The aims of this chapter are to:

1. Obtain information about the behaviour of particle distributions when a target experiences occlusion or camouflage. This is achieved by clustering particles at each time step. To provide a rich and widely applicable description of particle behaviour, a Gaussian Mixture Model is implemented to perform particle clustering.
2. Exploit information gathered from particle clustering to identify cues related to occlusion and camouflage. This is achieved using process-behaviour charts.

3.3. Particle clustering

3.3.1. Gaussian Mixture Model and Expectation Maximization (EM) algorithm

During the preliminary stages of the research, it was not clear what features/parameters of the particle set were going to be important. EM provides a rich representation (a Gaussian Mixture Model) which does not just assign data points to clusters. EM also gives probabilities of clusters membership and generates component weights summarising the importance of each cluster. Though in its basic form it needs the user to input the number of clusters to find, there exist extensions that can automatically determine the optimal number of clusters. For this reason, EM and GMM were chosen instead of the simpler K-Means (MacQueen, 1967, Steinhaus, 1956) and Mean Shift (Fukunaga et al., 1975) clustering algorithm. EM and Gaussian Mixture Models have previously been used successfully to represent particle sets in (Milstein et al., 2002) and (Vermaak et al., 2003).

A Gaussian Mixture Model is a probabilistic model used for density estimation in pattern recognition (Bishop, 1995). Probabilistic models can handle noise, capture unlikely but logically possible events and support a variety of principled reasoning methods. Gaussian Mixture Model parameters are often estimated using the Expectation Maximization (EM) algorithm.

The EM algorithm is an iterative maximum likelihood estimation process, mainly used when the data of interest are incomplete. The EM algorithm was formalized by (Dempster et al., 1977). Two main applications of the EM algorithm exist. The first arises when the data has missing values due to problems with or limitations of the observation process; and the second is when optimizing the likelihood function is analytically intractable but when the likelihood function can be simplified by assuming the existence of values for additional but missing (or hidden) parameters (Bilmes, 1997). The work presented here falls into the former class.

The EM algorithm iterates between two steps, the Expectation step and the Maximization step. The Expectation step finds the expected value of the log likelihood $\log p(X,Y \mid \Theta)$ with respect to the unidentified data Y given the observed data X and the current parameter estimation (we follow the notations in (Bilmes, 1997)):

$$Q(\Theta,\Theta^{(i-1)}) = E[\log p(X,Y \mid \Theta) \mid X,\Theta^{(i-1)}] \tag{3.1}$$

where $\Theta^{(i-1)}$ is the current parameter estimation that is used to evaluate the expectation and Θ is the new parameter that is optimized to increase Q. The Maximization step then finds an improved parameter estimation by maximizing the expectation function in the first step:

$$\Theta^{(i)} = \arg\max Q(\Theta,\Theta^{(i-1)}) \tag{3.2}$$

At each iteration, the log-likelihood increases. As a result, when a specified criterion is met, the algorithm stops iterating as it has converged to a local maximum of the log-likelihood function.

Literature shows the EM algorithm to be widely applicable in numerous fields including medical image analysis (Bromiley et al., 2003, Ziyan et al., 2009), pattern recognition and pattern classification in artificial intelligence (Kim et al., 2006, Cheung et al., 2002, Figueiredo, 2003, Lim et al., 2005, Williams et al., 2007).

3.3.2. Modelling particle distribution via a Gaussian Mixture Model and EM

In this thesis, an iterative EM algorithm is used to build a Gaussian Mixture Model to describe the particle set present in a Condensation tracker at each time step. (Redner et al., 1984, Xu et al., 1996) shows that there are a number of advantages in using the EM algorithm to build mixture models. Among the advantages are reliable global convergence, low cost per iteration, economy of storage, ease of programming, automatic satisfaction of probabilistic constraint and monotonic convergence without the need to set a learning rate.

Clustering of particles is not the norm in visual tracking, but has been done as shown in (Milstein et al., 2002) and (Vermaak et al., 2003). In (Vermaak et al., 2003), the EM algorithm is used as a clustering tool to cluster together particles from one time frame. Cluster membership is taken into account when particles are propagated into the next time frame.

This thesis approach differs from the work reported in (Milstein et al., 2002) and (Vermaak et al., 2003) in a number of ways. Here, cluster membership is less important than the global properties (or parameters) of the clusters produced. Moreover, clusters are not propagated forward in time, but are recalculated at each time step. Particles do not stay within the same cluster throughout the tracking process, and clusters may vary dramatically in size. The details of the Gaussian Mixture Model algorithm can be found in section A.1 of this thesis.

The functionality of the EM algorithm is governed by three parameters: a convergence criterion, an upper limit on the number of iterations performed and the number of clusters utilised. The following section examines and justifies the values chosen for each.

3.3.2.1. Number of iterations

The maximum iteration for the EM algorithm is set at 50. Analysis shows that the EM algorithm successfully converged even before reaching the maximum limit. The EM algorithm converges early, without having to iterate to the maximum iteration limit even when using different number of clusters. As a result, stopping the EM algorithm after iterating a maximum of 50 iterations is justifiable as choosing a higher iteration limit will cause the EM algorithm to iterate uneconomically and utilise additional computation time. Analysis also showed that the number of iteration of the EM algorithm does not provide any reliable effect to indicate the occurrence of occlusion or camouflage.

3.3.2.2. Convergence Criterion

The convergence criterion used within EM plays an important role in monitoring the progress of the algorithm. EM is implemented as an iterative process which continues to iterate until all clusters are stable. A convergence criterion immediately stops the EM algorithm once the clusters have stabilised, thus avoiding additional and unnecessary iterations.

The convergence of the EM algorithm is measured by calculating the Euclidean distance between the cluster's mean (x, y) coordinates in the current and previous iterations. The process is considered to have converged when this distance falls below an empirically determined convergence threshold. During occlusion, the EM algorithm often fails to converge and instead performs the maximum allowed number of iterations. The reason for this is that the particle set often moves beyond the boundary of the image plane after losing its target, making effective clustering of these particles impossible. Analysis also provides evidence that the Euclidean distance and convergence threshold are poor indicators of occlusion and camouflage.

3.3.2.3. Number of clusters

To determine the number of clusters that should be considered EM was applied to the particle sets obtained from a Condensation algorithm processing the set of test sequences presented in Chapter 2. The number of clusters used was 2, 3, and 4. Selected results are included here; the remainder are documented in Appendix B.

Two additional artificial test videos designed to exhibit multi modal particle sets were also created. In these, a differently (to invoke occlusion) or an identically (to invoke camouflage) coloured rectangle is placed in the path of the moving yellow circle. An additional two yellow circles are stationed prior to the interfering object to encourage multi-modality. Initially, the two yellow circles are stationary. Once the moving yellow circle interacts with the two stationary circles, all three move towards the interfering object. The two sequences are illustrated in Figure 3.1 and Figure 3.2 respectively.

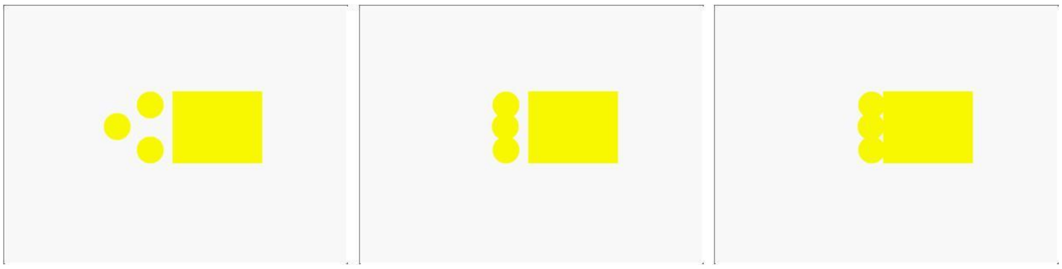


Figure 3.1 Camouflage of a multi modal particle set.

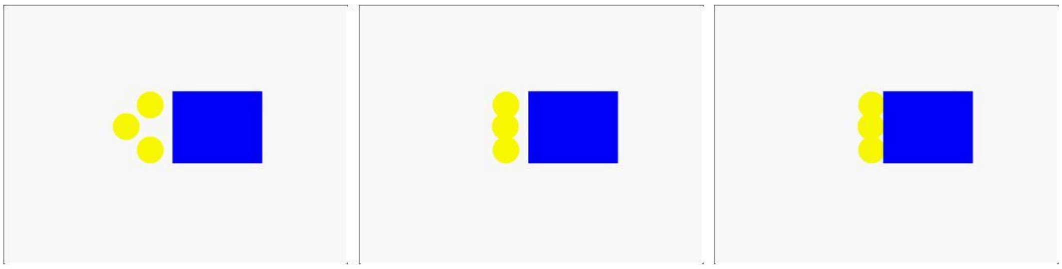


Figure 3.2 Occlusion of a multi modal particle set.

Figure 3.3, Figure 3.5 and Figure 3.7 show the clusters found by EM during the occurrence of camouflage, albeit the use of 2, 3 or 4 clusters at the start of tracking. Analysis of graphs in Figure 3.4, Figure 3.6 and Figure 3.8 shows that regardless of the number of clusters EM is required to find, it usually only finds one or two clusters. This is because the target is very simple and can be described by one or two clusters. More importantly, there is no systematic change in the number of clusters, reported in the graph as the target becomes camouflaged.

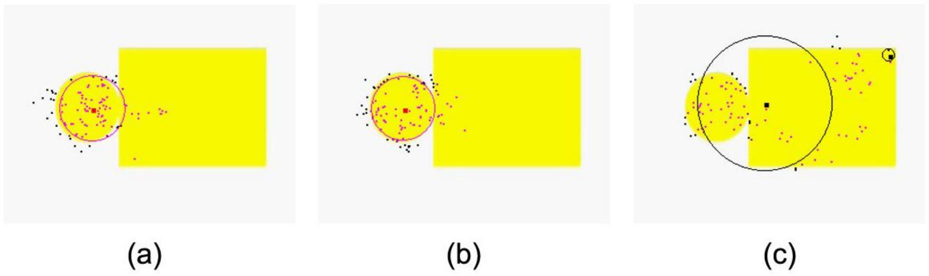


Figure 3.3 A yellow circle experiencing camouflage when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

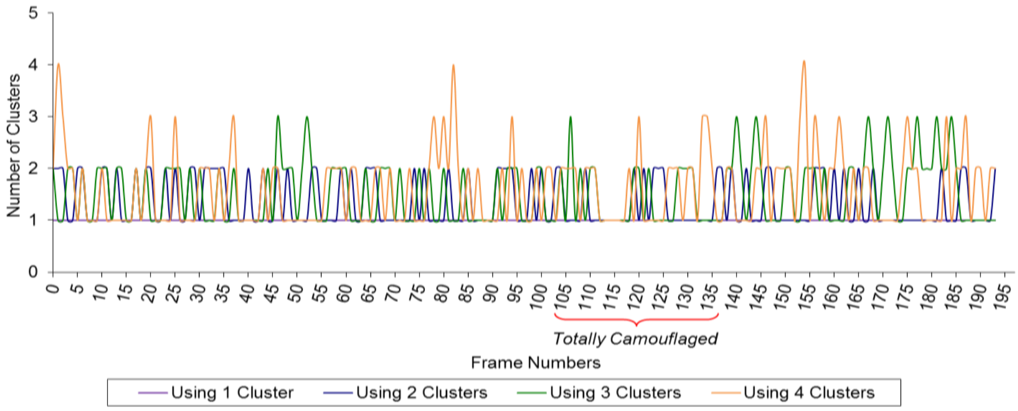


Figure 3.4 Graph showing the use of different numbers of clusters for a yellow circle experiencing camouflage.

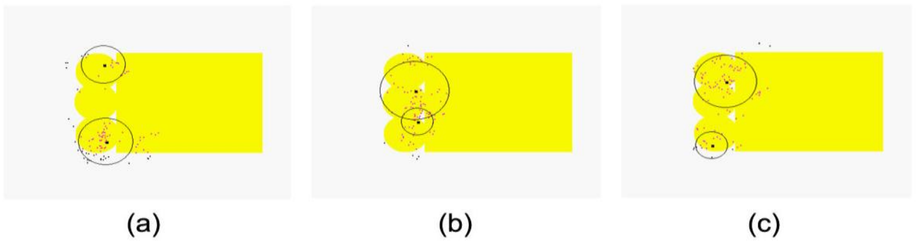


Figure 3.5 Tracking a camouflaged object with a multi-model particle set when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

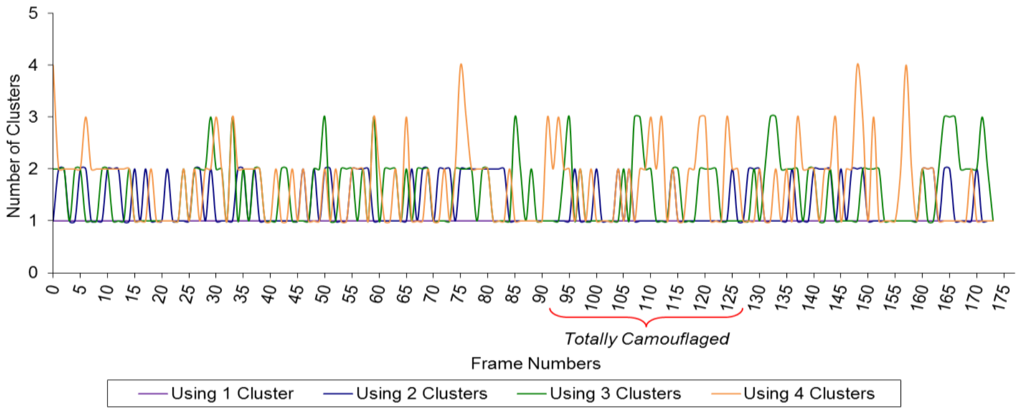


Figure 3.6 Graph showing the use of different numbers of clusters when tracking a camouflaged object with a multi-modal particle set.

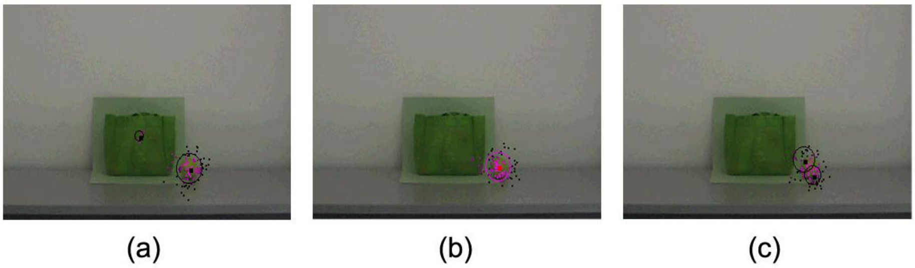


Figure 3.7 A tennis ball experiencing camouflage when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

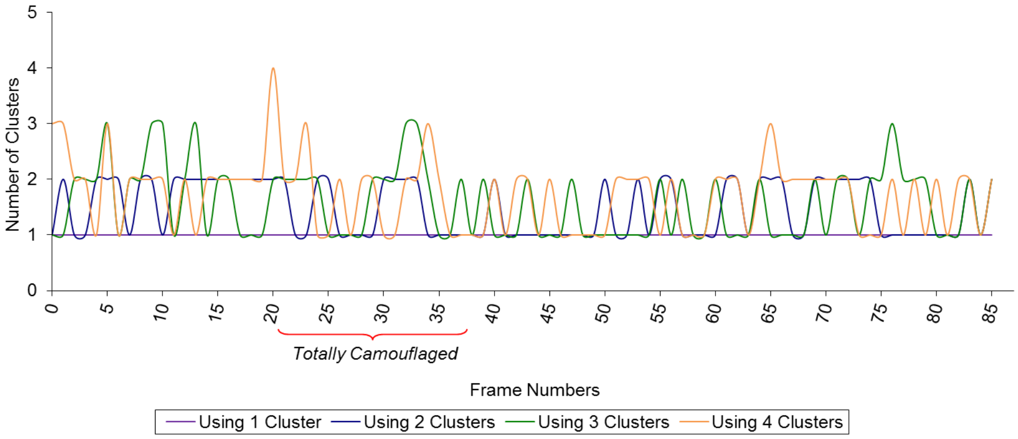


Figure 3.8 Graph showing the use of different numbers of clusters for a tennis ball experiencing camouflage.

Figure 3.9, Figure 3.11 and Figure 3.13 shows the clusters found by EM during the occurrence of occlusion, even though at the start of tracking, the number of clusters was initialised to 2, 3 or 4 clusters. Analysis of graphs in Figure 3.10 and Figure 3.14 shows that there is an increase in clusters as the targets become occluded, as the particles bunch together on the occluding object. However, it is not a predictable event, as it is a function of the difference between the two objects, and not reliable enough to base a detection method on. Additionally, in Figure 3.12, the number of clusters goes up a bit earlier. This implies that there could be an increase in cluster numbers because there are similar objects nearby or because the target is occluded. So, while there might be an increase in cluster numbers when during occlusion, again, it is dangerous to rely on it.

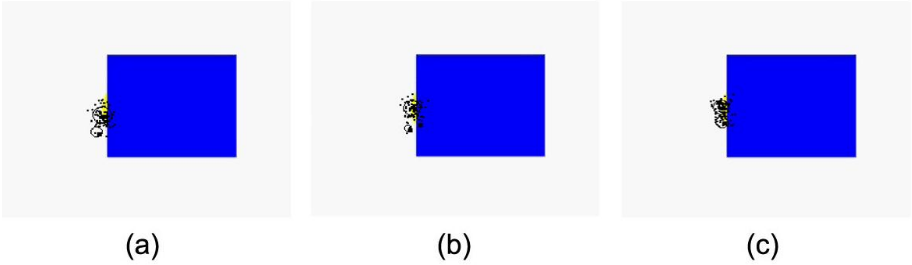


Figure 3.9 A yellow circle experiencing occlusion when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

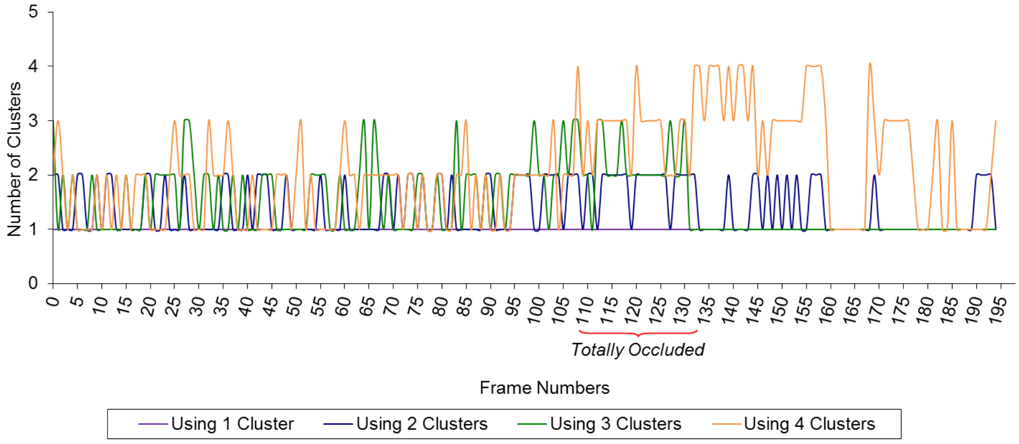


Figure 3.10 Graph showing the use of different numbers of clusters for a yellow circle experiencing occlusion.

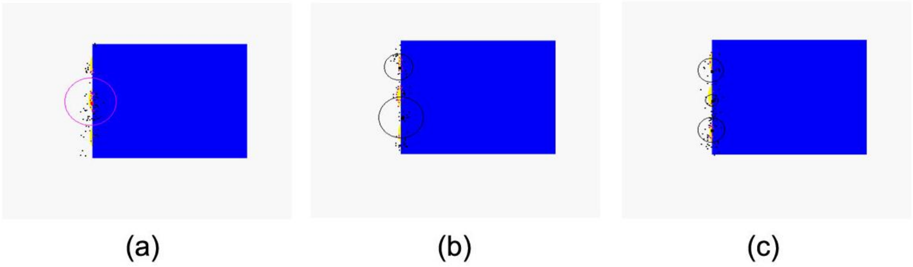


Figure 3.11 Tracking an occluded object with a multi-model particle set when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

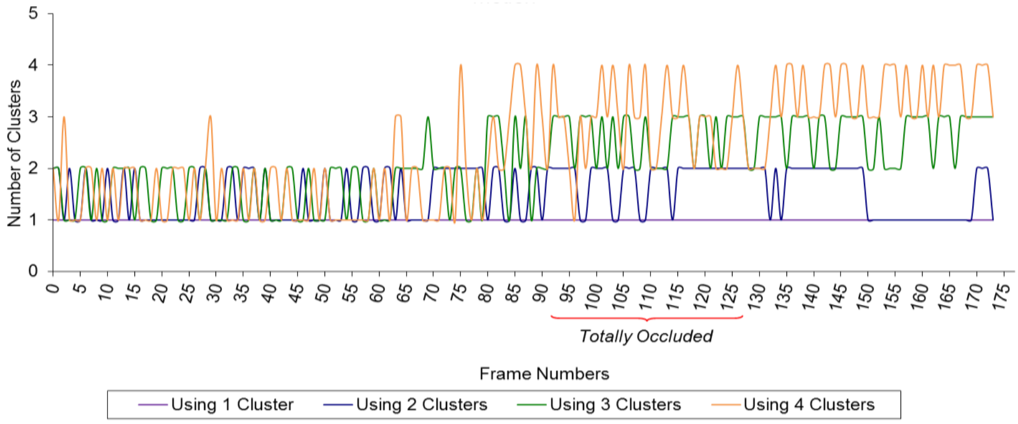


Figure 3.12 Graph showing the use of different numbers of clusters when tracking an object through occlusion with a multi-model particle set.

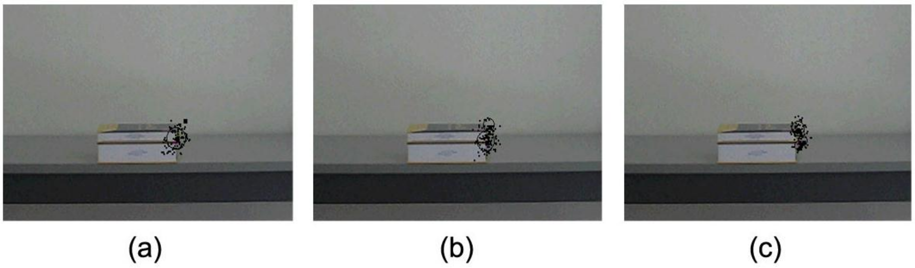


Figure 3.13 A tennis ball experiencing occlusion when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

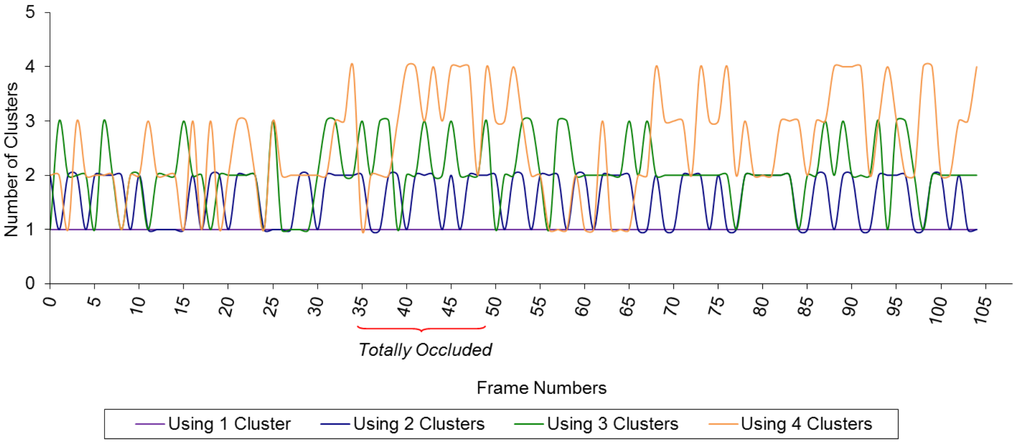


Figure 3.14 Graph showing the use of different numbers of clusters for a tennis ball experiencing occlusion.

Analysis provides evidence that, regardless of the number of distinct clusters present in the particle set, the number of clusters does not change significantly when the target is occluded or camouflaged.

While there are some variations in cluster number during occlusion and camouflage when EM is initialised with 2, 3 or 4 clusters, these variations are not robust enough to provide a strong and reliable cue indicating occlusion or camouflage. A single cluster will therefore be used to provide information about particle set behaviour during occlusion and camouflage. Though EM was adopted here as a valuable experimental tool, this result means that its use in any applied system is strictly unnecessary; if the particle set can reliably be assumed to form a single cluster, its mean position and deviation can be computed directly. The EM implementation described here is, however, employed, for consistency, throughout the remainder of this thesis.

3.3.3. Discussion

The three parameters: (1) an upper limit on the number of iterations performed; (2) convergence criterion and (3) the number of clusters utilised, play a pivotal role in fine-tuning the EM algorithm. However, none of these parameters can be used to indicate occlusion or camouflage as neither of the effects which takes place when these events occurs provide a strong and reliable indicator to the occurrence of these events.

3.4. Camouflage, occlusion and particle cluster properties

Fitting a Gaussian Mixture Model to the particle set provides information regarding the collective particle behaviour while tracking a target moving normally, or during occlusion or camouflage. Results reported in section 3.3 suggest that, while number of clusters, number of iterations and convergence behaviour show some variation when camouflage and occlusion takes place, the effects are not strong and cannot reliably be used as an indicator of these events.

In what follows the particle set is modelled as a single cluster, with attention focussed on the properties of that cluster. At each time-step, the EM algorithm estimates the (mean) position and spread (deviation) of a single particle cluster. The tracker reports estimated speed, and the motion model used to project the particles forward is also recorded, allowing comparison of expected and actual speed. This information is plotted for the entire image sequence and displayed graphically. The particle cluster mean is shown as a pink dot and the particle cluster deviation is shown as a pink circle overlaid on each image.

This data has been extracted from the test videos described in Chapter 2. The results are discussed in this section. All graph results are average graphs obtained after running the Gaussian Mixture Model algorithm on each test videos ten times.

The Gaussian Mixture Model algorithm is a point based implementation. The initial starting point for each sequence is user defined. The same parameter settings used for the Condensation analysis in Chapter 2 are used for the particle cluster analysis in this chapter. In all test videos, a particle set size of 100 is used. The degree of process noise added varied for each test video depending on the target size. A radius size ranging from 1 to 10 is selected to produce a good tracking result. A histogram bin size of 10 is selected for all the test videos.

3.4.1. Normal tracking

Figure 3.15, Figure 3.17 and Figure 3.19 show the particle cluster data describing Condensation particles which successfully track the target (a yellow circle and a tennis ball) from the start of the image sequence to the end. The graphs showing the average particle cluster data for these 3 test videos are plotted in Figure 3.16, Figure 3.18 and Figure 3.20.

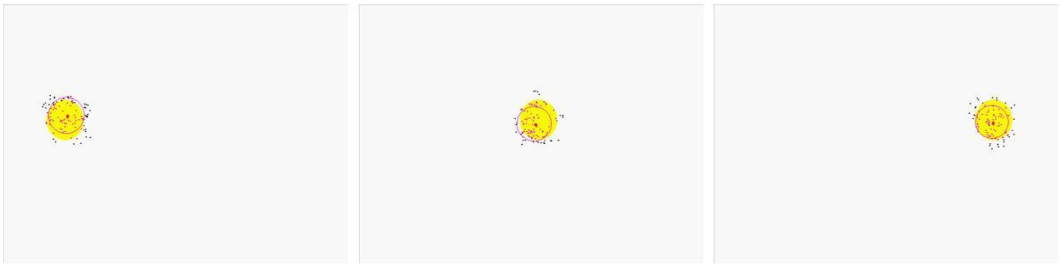


Figure 3.15 Particle cluster data describing Condensation particles tracking a yellow circle moving normally.

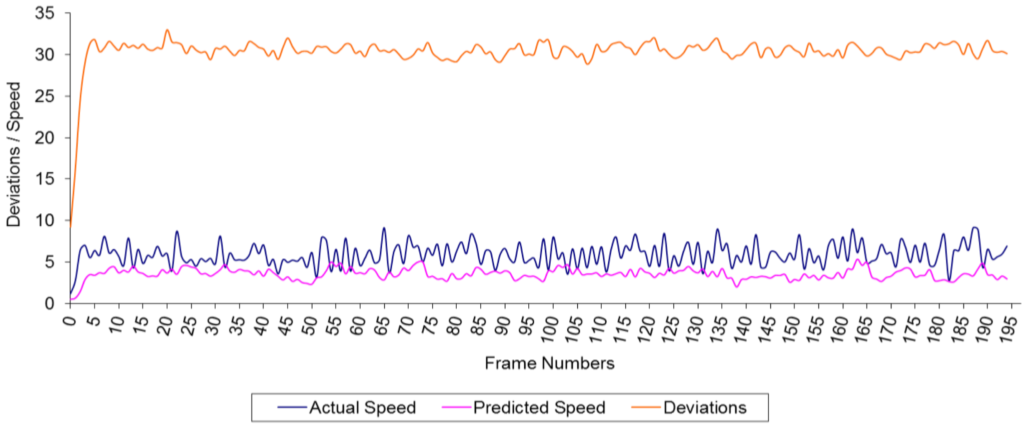


Figure 3.16 Graph showing the average particle cluster data for a yellow circle moving normally.

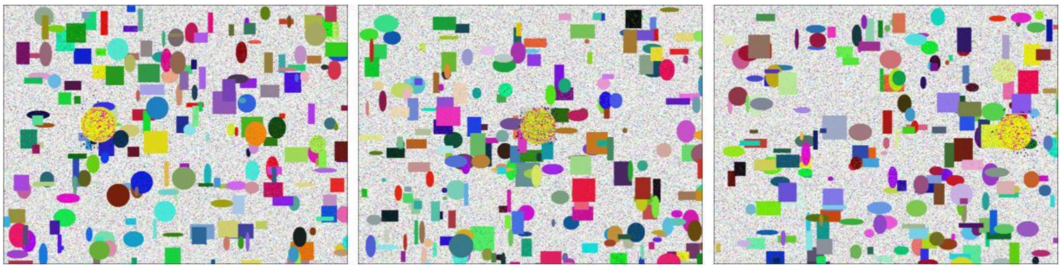


Figure 3.17 Particle cluster data describing Condensation particles tracking a yellow circle moving normally amid clutter.

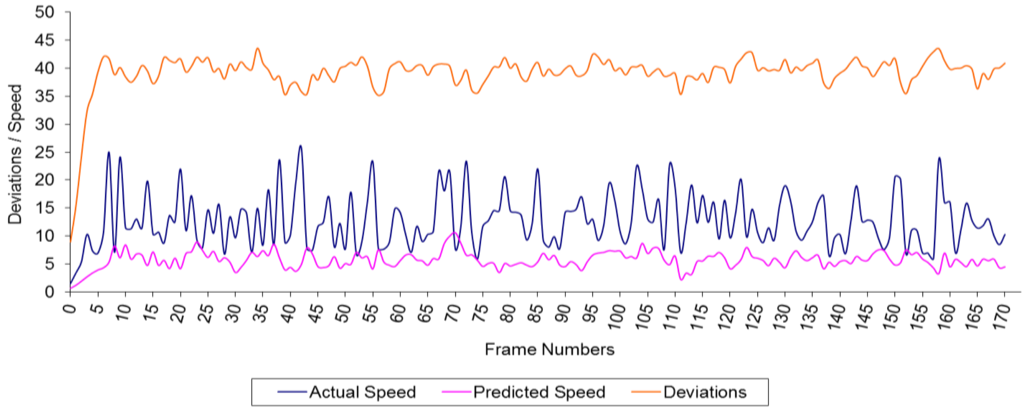


Figure 3.18 Graph showing the average particle cluster data for a yellow circle moving normally amid the presence of clutter.

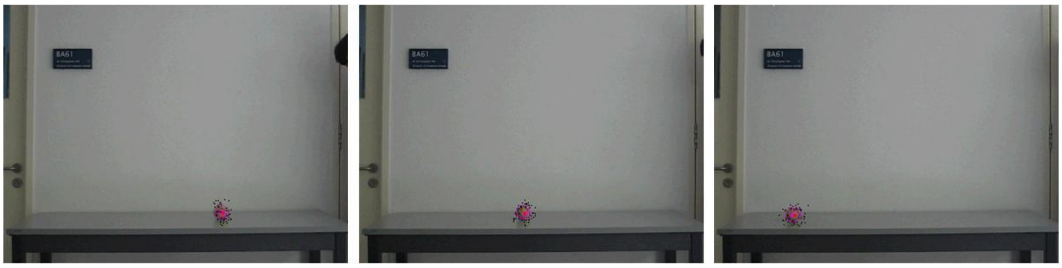


Figure 3.19 Particle cluster describing Condensation particles tracking a tennis ball moving normally.

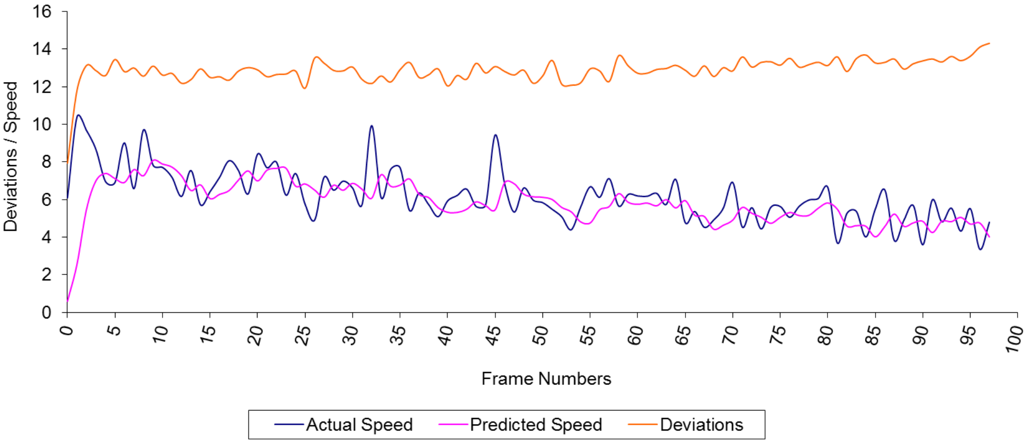


Figure 3.20 Graph showing the average particle cluster data for a tennis ball moving normally.

Analysis of graphs shows that:

- The deviation remains approximately constant throughout, though there appears to be increased variation in the presence of clutter.
- Predicted and estimated speed is well-correlated.

3.4.2. Occlusion

Figure 3.21, Figure 3.23, Figure 3.25 and Figure 3.27 show the particle cluster data describing Condensation particles when the target (a yellow circle, a woman and a tennis ball) experiences occlusion. The average data values for each of these test videos are plotted in Figure 3.22, Figure 3.24, Figure 3.26 and Figure 3.28. Figure 3.29 demonstrates the Gaussian cluster properties describing Condensation particles when the target (a table tennis bat) experiences self-occlusion, while the average data values are plotted in Figure 3.30.

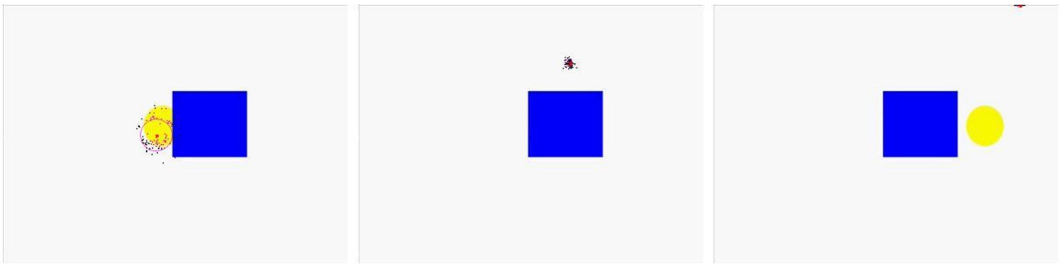


Figure 3.21 Particle cluster data describing Condensation particles tracking a yellow circle experiencing occlusion.

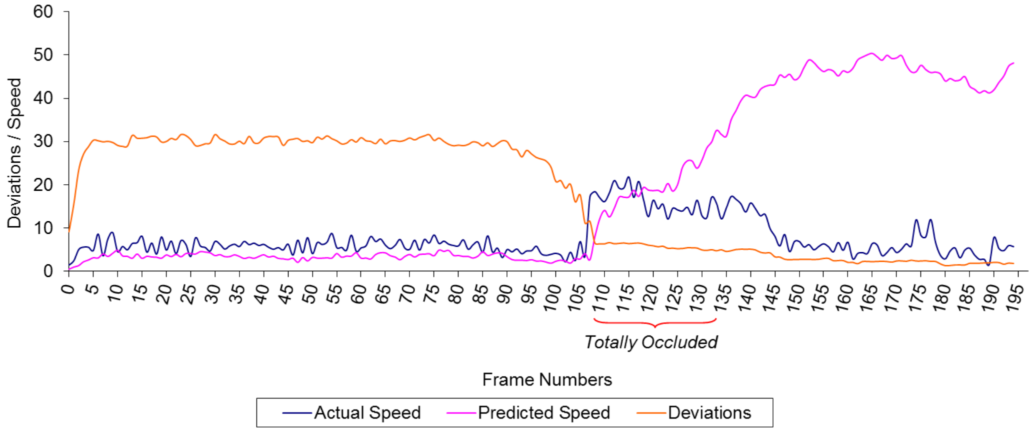


Figure 3.22 Graph showing the average particle cluster data obtained from a yellow circle experiencing occlusion.

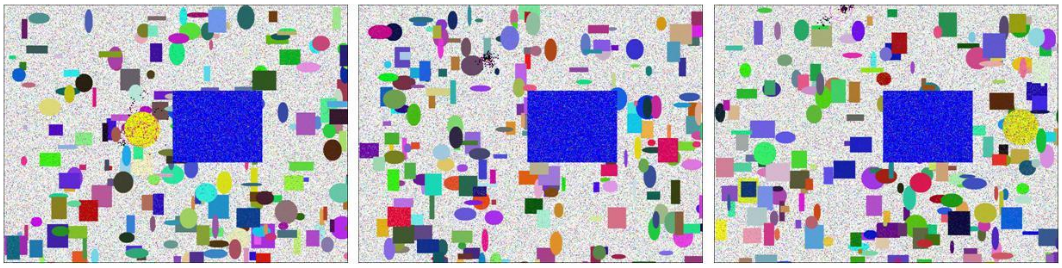


Figure 3.23 Particle cluster data describing Condensation particles tracking a yellow circle experiencing occlusion amid the presence of clutter.

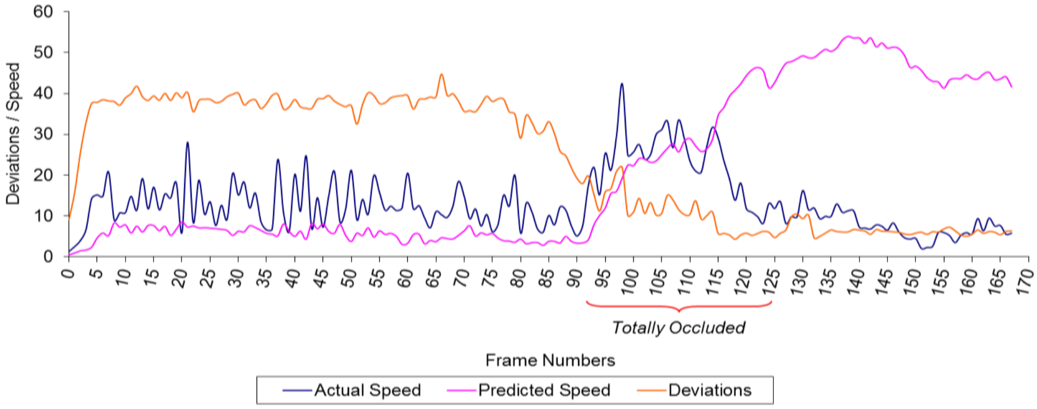


Figure 3.24 Graph showing the average particle cluster data (over 10 trials) for a yellow circle experiencing occlusion amid clutter.

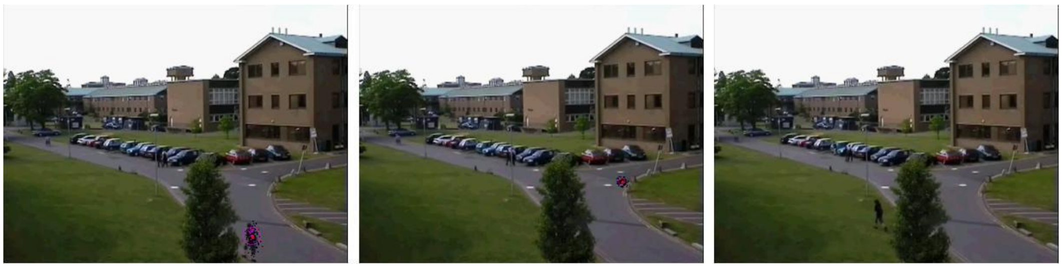


Figure 3.25 Particle cluster describing Condensation particles tracking a woman as she becomes occluded by a tree.

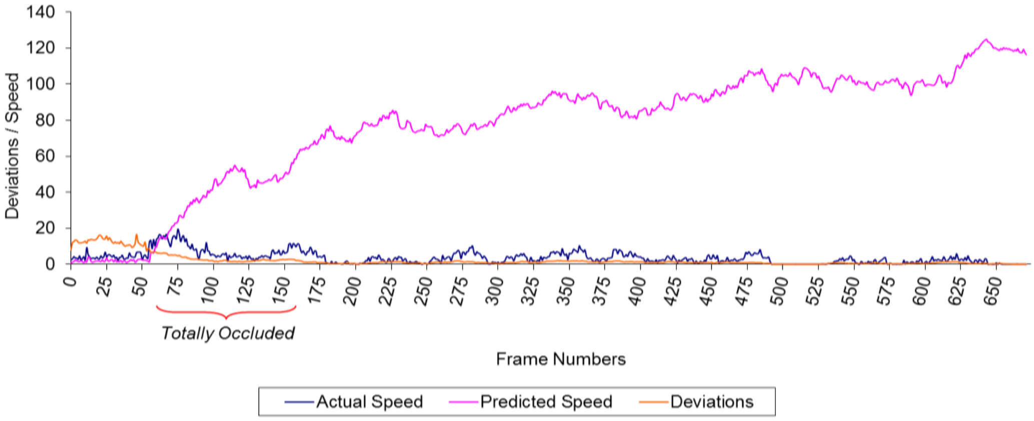


Figure 3.26 Graph showing the average cluster data for tracking a woman becoming occluded by a tree.

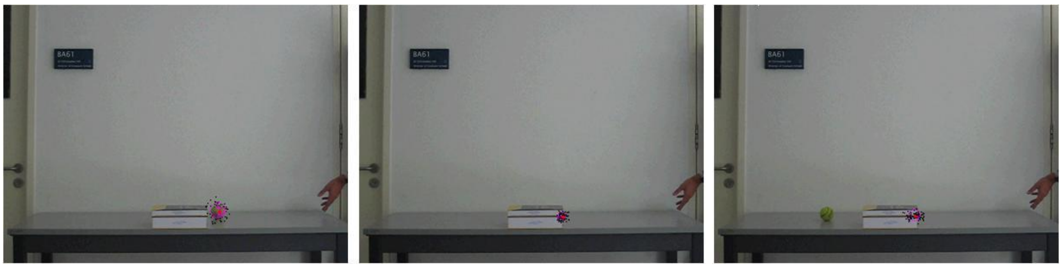


Figure 3.27 Particle cluster describing Condensation particles tracking a tennis ball experiencing occlusion.

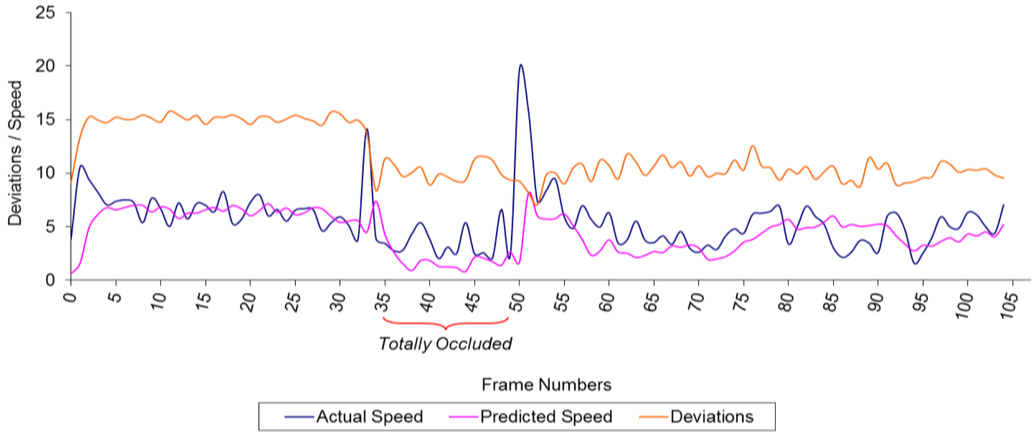


Figure 3.28 Graph showing the average cluster data for a tennis ball experiencing occlusion.

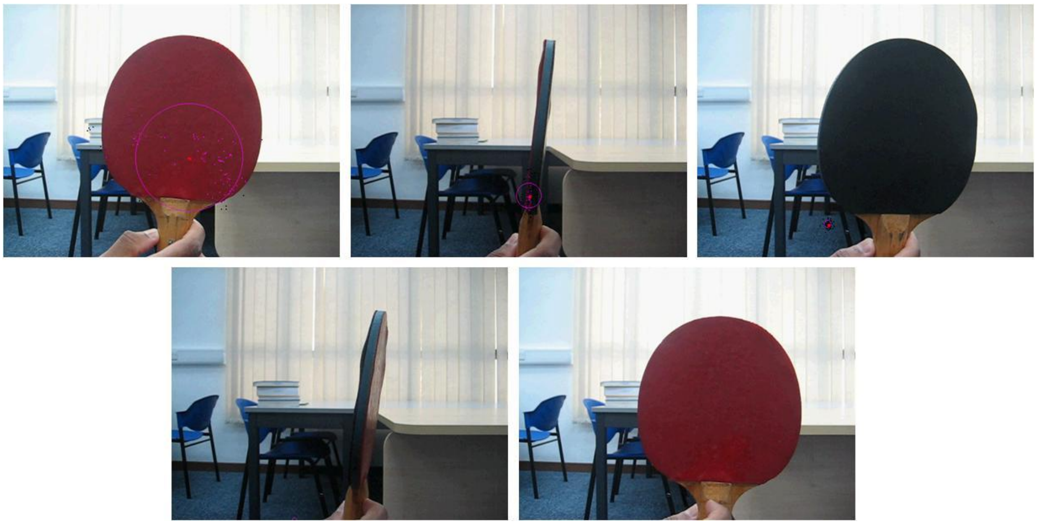


Figure 3.29 Particle cluster describing Condensation particles tracking a table tennis bat experiencing self-occlusion.

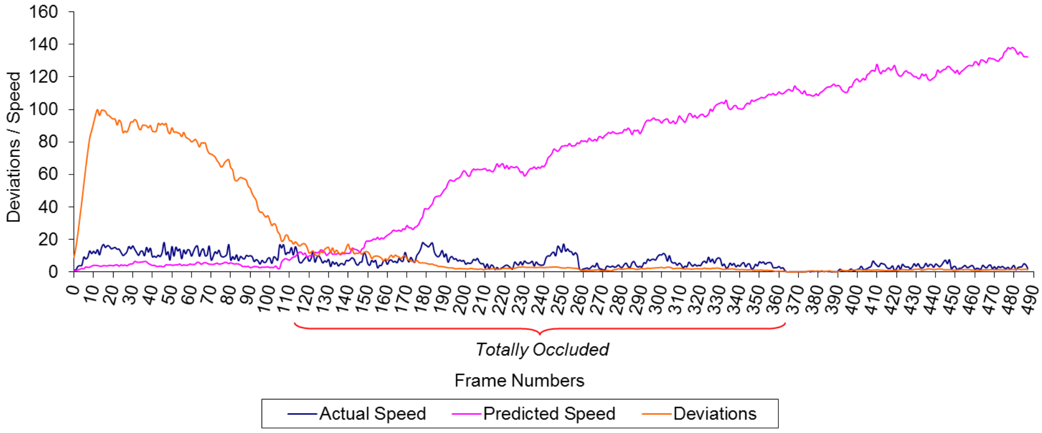


Figure 3.30 Graph showing the average cluster data for a table tennis bat experiencing self-occlusion.

As the target becomes occluded:

- The particle deviation (spread) experiences a systematic drop. This is due to the particles becoming more tightly clustered on the shrinking visible portion of the target (prior to the target being fully occluded).
- This is commonly accompanied by a corresponding increase in the predicted and estimated speed. The sudden increase in speed is the result of the particles losing the target and moving quickly across the occluding object, prior to being destroyed.

3.4.3. Camouflage

The particle cluster data obtained from Condensation particles when the target (a yellow circle, a football and a tennis ball) experiences camouflage is shown in Figure 3.31, Figure 3.33, Figure 3.35 and Figure 3.37. The average cluster data plot for each of these test sequences is shown in Figure 3.32, Figure 3.34, Figure 3.36 and Figure 3.38.

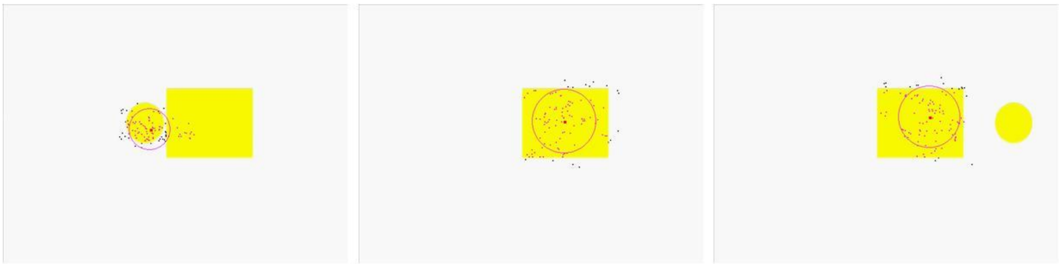


Figure 3.31 Cluster data describing Condensation particles tracking a yellow circle experiencing camouflage

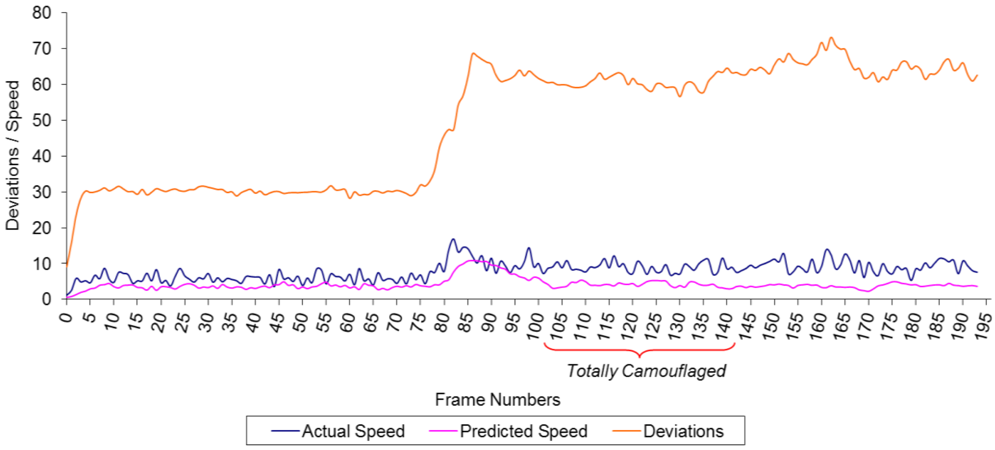


Figure 3.32 Graph showing the average particle cluster data for a yellow circle experiencing camouflage.

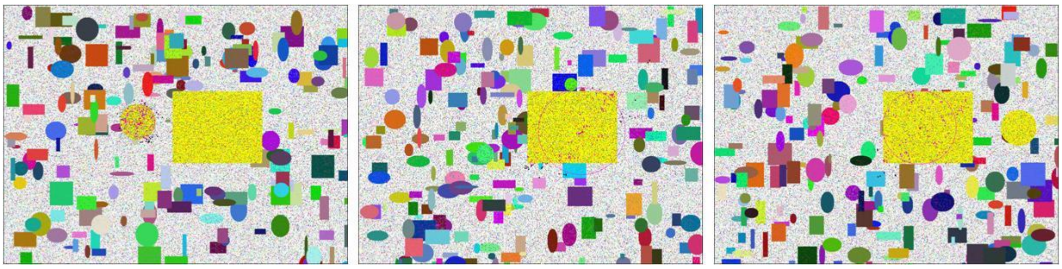


Figure 3.33 Particle cluster describing Condensation particles tracking a yellow circle experiencing camouflage amid clutter.

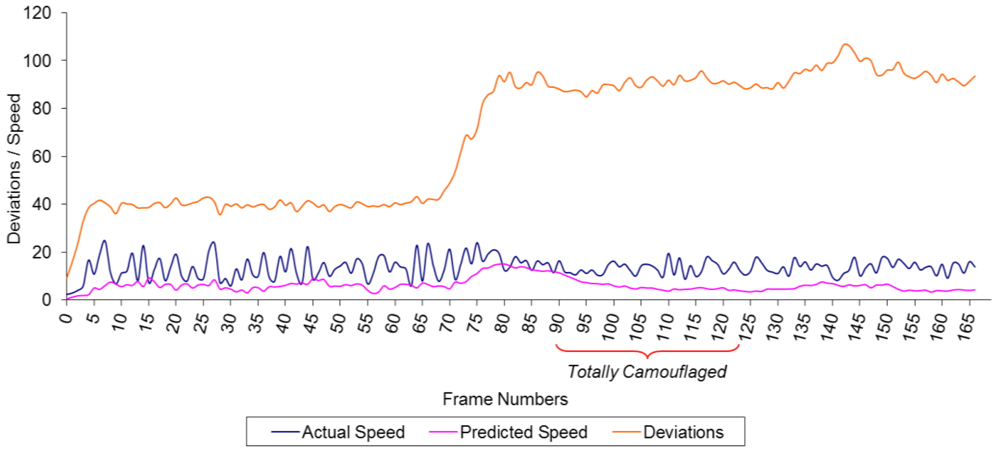


Figure 3.34 Graph showing the average particle cluster data for a yellow circle experiencing camouflage amid clutter.

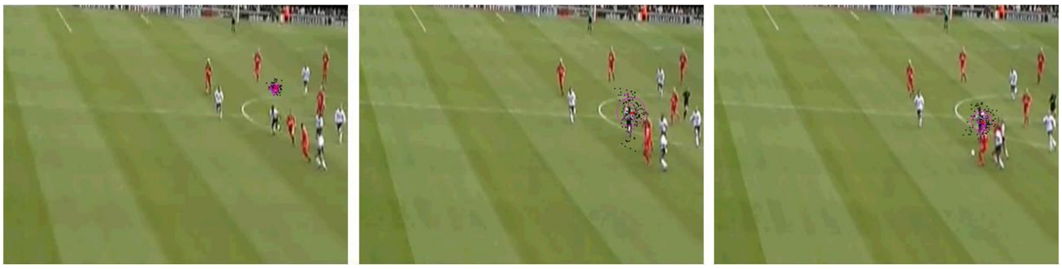


Figure 3.35 Particle cluster describing Condensation particles tracking a football as it becomes camouflaged by a player.

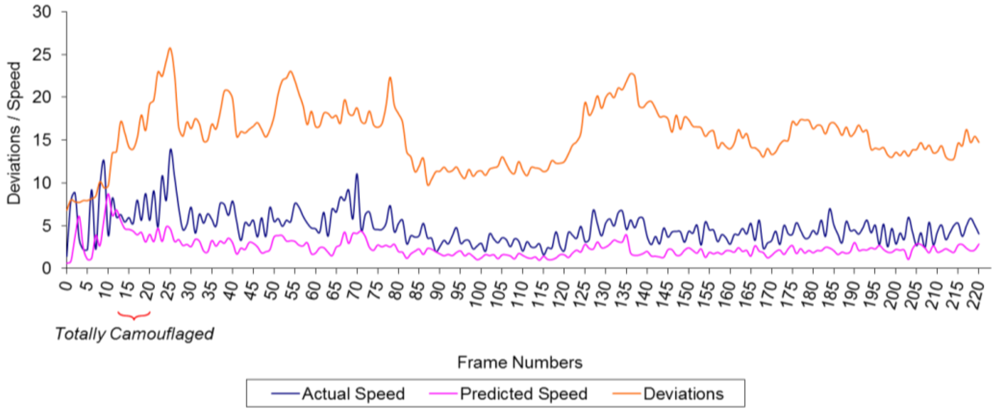


Figure 3.36 Graph showing the average particle cluster data for a football being camouflaged by a player.

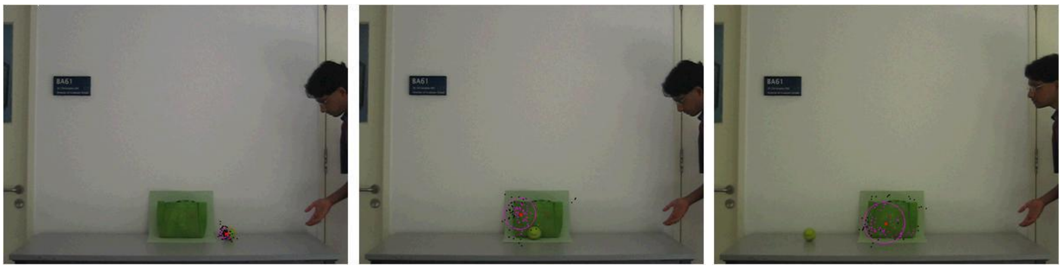


Figure 3.37 Particle cluster describing Condensation particles tracking a tennis ball experiencing camouflage.

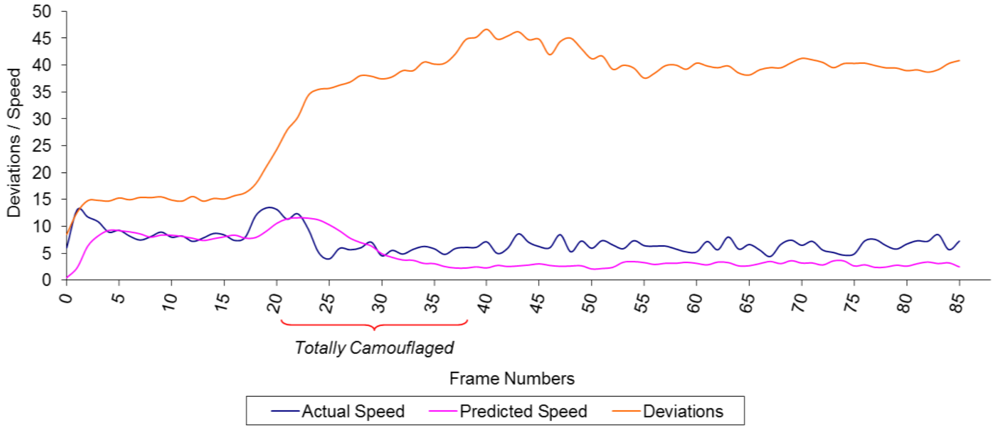


Figure 3.38 Graph showing the average cluster data for a tennis ball experiencing camouflage.

As the target becomes camouflaged, the graphs show that:

- The particle deviation (spread) increases. The systematic increase is caused by the transfer of particles spreading out across the (larger) camouflaging object.
- The transfer and spread out of particles across the camouflage object also increases the estimated (and so the predicted) speed as well.

3.4.4. Summary

Analysing the parameters of the particle cluster produced by the EM algorithm leads to the following summary:

- During normal tracking, the cluster deviation remains approximately constant, while the speed is also well-correlated.
- During occlusion, the cluster deviation drops but the speed of the particles within the cluster increases. The reduction in the deviation is due to the particles being clustered at the (still visible) back of the target. The increase in speed occurs because during occlusion, a small number of particles will land on the occluding object, move quickly across it and then be destroyed as their weight drops to zero.
- During camouflage, the cluster deviation increases, as does the speed of the particles within the cluster. The increase in deviation and speed can be attributed to a general expansion of the cluster when the particles are transferred onto the similar, but larger, camouflaging object.

As a result, the ability to reliably detect these changes in the particle set may enable a particle filter-based tracker to automatically identify occlusion and camouflage events. This is explored in section 3.5.

3.4.5. 2D EM Algorithm VS 3D EM Algorithm

Throughout this chapter, the EM algorithm implemented to build the Gaussian Mixture Model exploits only the position of the particle. The Condensation weights are not taken into consideration. The Condensation weight, Π provide the tracker with information as to which particles are most likely to remain on the target and which ones are most likely to be propagated into the next frame. This extra information can be incorporated into the occlusion/camouflage detection method described in section 3.5 by simply extending the EM algorithm from two to three dimensions. A revised mathematical representation for a three dimensional EM algorithm is detail in section A.2.

Figure 3.39 to Figure 3.44 are graphs showing the average particle cluster data, generated after running the two-dimension EM algorithm and the three-dimension EM algorithm onto artificial videos with clutter present for a total of ten times each. Figure 3.39 and Figure 3.40 shows the graph of using a two-dimension EM algorithm clustering and a three-dimension EM algorithm clustering while tracking a target moving normally. The graphs using two-dimension EM algorithm clustering and three-dimension EM algorithm clustering while tracking a target experiencing occlusion are shown in Figure 3.41 and Figure 3.42 respectively. The graphs illustrated in Figure 3.43 and Figure 3.44 shows the use of a two-dimension EM algorithm clustering and a three-dimension EM algorithm clustering while tracking a target experiencing camouflage.

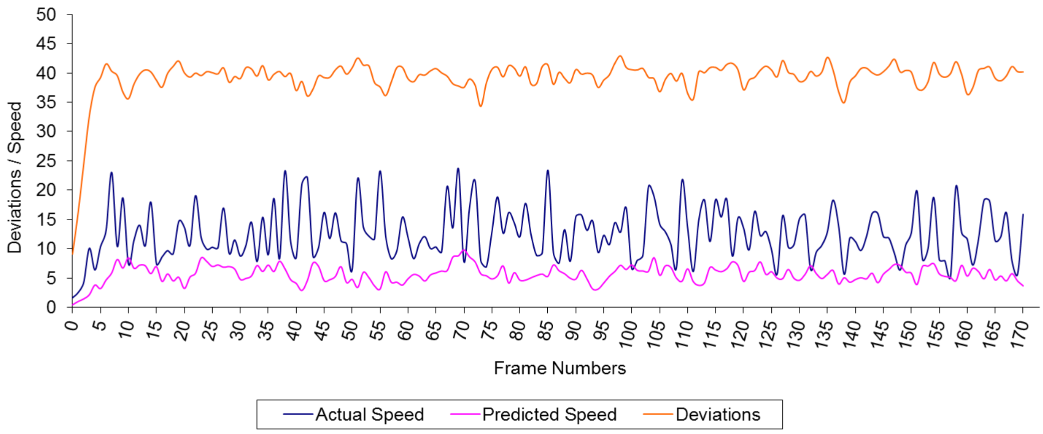


Figure 3.39 Graph showing the use of 2D EM-algorithm clustering while tracking normally.

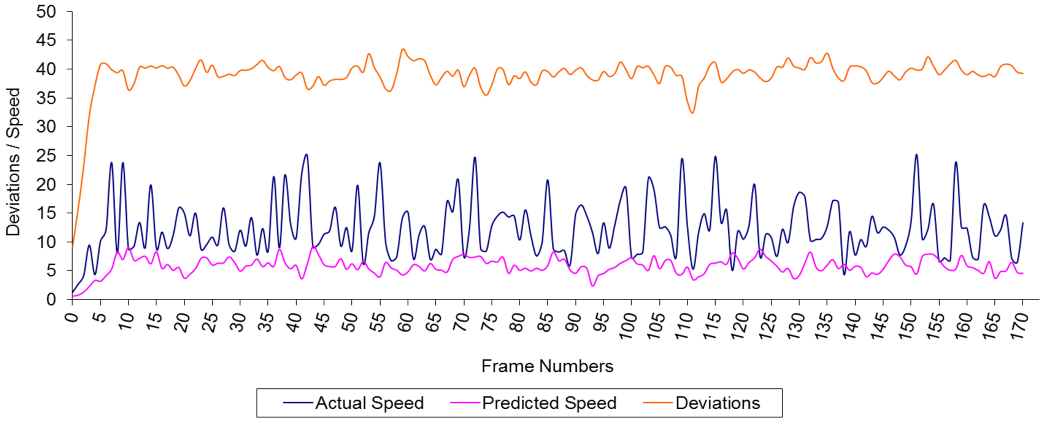


Figure 3.40 Graph showing the use of 3D EM-algorithm clustering while tracking normally.

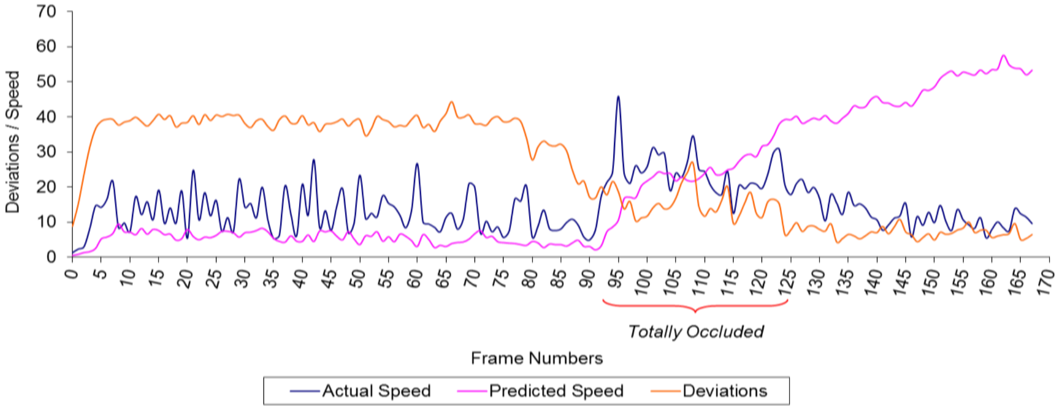


Figure 3.41 Graph showing the use of 2D EM-algorithm clustering while experiencing occlusion.

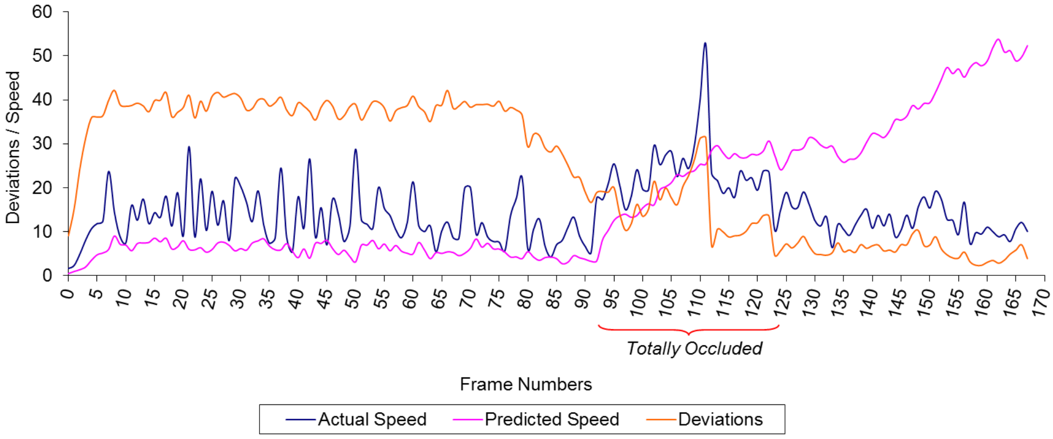


Figure 3.42 Graph showing the use of 3D EM-algorithm clustering while experiencing occlusion.

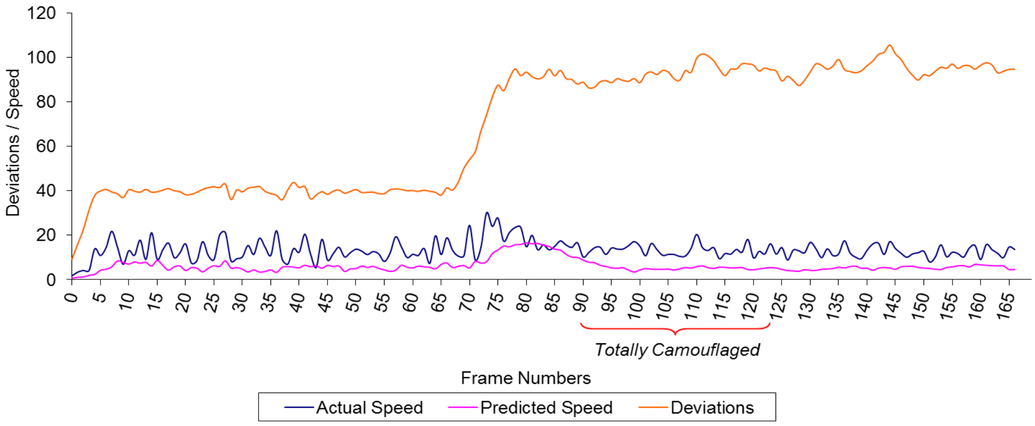


Figure 3.43 Graph showing the use of 2D EM-algorithm clustering while experiencing camouflage.

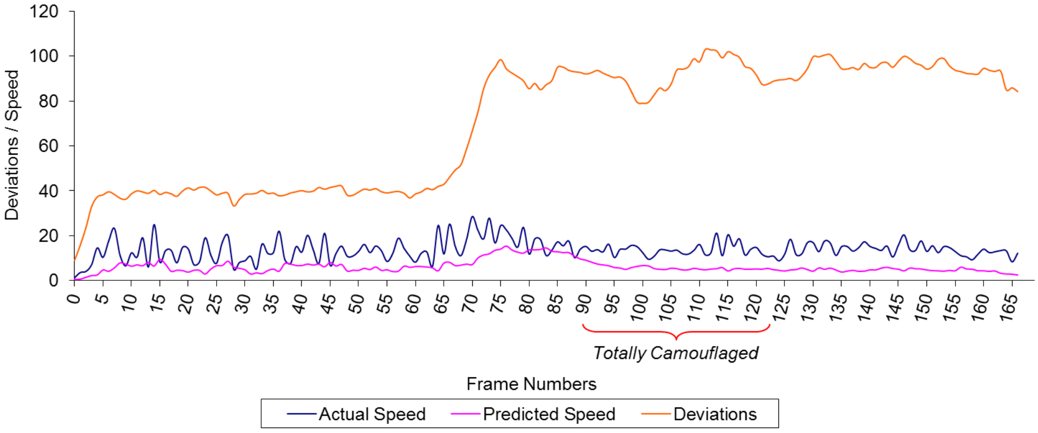


Figure 3.44 Graph showing the use of 3D EM-algorithm clustering while experiencing camouflage.

Analysis of the graphs using a two-dimension EM algorithm and a three-dimension EM algorithm on artificial videos with clutter present demonstrates that the clustered particle deviation and speed undergoes no significant changes with the inclusion of the Condensation weights. Hence, the decision to implement a two-dimensional EM algorithm for this thesis is justifiable.

3.5. Detecting occlusion and camouflage

Analysis of particle cluster data provides evidence that occlusion and camouflage have a direct affect on the particle cluster deviation (spread) and particle cluster speed. Hence, we hypothesise that by examining the fluctuating behaviour of the particle cluster spread and particle cluster speed during tracking, the occurrence of occlusion and camouflage can be identified.

The current state of these parameters is compared with a model representation. Each parameter has an independent model representation associated with it. These model representations are created during the initial period of tracking when the target is known or assumed to be clearly visible. Comparison of data and model is achieved via process behaviour charts.

3.5.1. Process-behaviour charts

Process-behaviour charts, also known as control charts, are graphical statistical tools used to monitor the behaviour of fluctuating data. Literature review has shown that process-behaviour charts are primarily a control engineering tool and have not previously been used in visual tracking.

The Shewhart (Shewhart, 1931) and Exponentially Weighted Moving Average (EWMA) (Roberts, 1959) process-behaviour charts are the most commonly used. Shewhart charts are typically used to detect large changes in the mean and variation of some data value. Whereas EWMA charts are used when small changes in data mean and variance must be detected (Zhang et al., 2009).

The EWMA algorithm relies on a user-specified parameter, λ , the choice of which is vital in making the chart sensitive to the changes sought. The value of λ must be between 0 and 1, with values nearing to 0 giving a priority to older data and values nearing to 1 giving priority to newer data. As determination of an appropriate λ is problematic, and particle set fluctuations are quite large, the Shewhart chart approach is employed here.

The Shewhart chart parameters are computed over a moving window using the cluster data obtained from the EM algorithm. The moving window interval information is not gathered during the first five frames of the video sequence as, during this stage, the particles are spreading within the boundary of the target object. Trial and error showed using a moving window size of 7 produced good results. Maximum and minimum control limits are calculated using (3.3) and (3.4):

$$MaximumControlLimit = \mu + 3\sigma \tag{3.3}$$

$$MinimumControlLimit = \mu - 3\sigma \tag{3.4}$$

On the chart, the maximum control limit is represented as *upper control limit* (UCL) while the minimum control limit is represented as *lower control limit* (LCL). These control limits are used as a measurement threshold to determine whether the data in question is still under control or has gone out of control. The maximum and minimum control limit points are computed from the mean of the data. The control limits used in this thesis is determined based on the Nelson rule (Nelson, 1984). In this thesis, rule 3 is implemented whereby, if six (or more) data points in a row continually increase (or decrease) beyond the control limits, then the process-behaviour chart will fire, indicating a significant change. Applying this rule handles false alarms such as data points that have exceeded the control limits for a short time period before returning to be under control, e.g. Figure 3.45.

In the event of the mean value exceeding the lower control limit, the tracker notifies the user that the target in question is experiencing occlusion. Should the upper control limit be exceeded, the target is considered to be experiencing camouflage.

```
PROCESS-BEHAVIOUR CHART ALGORITHM
1) Record particle cluster result obtained from implementing the EM algorithm.
2) Determine whether the model representation has been created:
   a) If no model representation is created, then build a model
      representation by computing the maximum and minimum control
      points from the computed mean, $\mu_1$ and standard deviation, $\sigma_1$ of the
      initial moving window interval:
         i.  $MaximimControlLimit = \mu_1 + 3\sigma_1$
         ii. $MinumumControlLimit = \mu_1 - 3\sigma_1$
   b) If a model representation is already created, then build the current
      state representation by computing the current state mean, $\mu_t$ of the
      moving window interval.
3) Compare current state mean, $\mu_t$ with the control limits:
   a) If the current state mean, $\mu_t$ stays within the boundary of the control
      limits, then the target is moving normally.
   b) If the current state mean, $\mu_t$ exceeds the maximum control limits, then
      the target is experiencing camouflage.
   c) If the current state mean, $\mu_t$ exceed the minimum control limits, then
      the target is experiencing occlusion.
```

Algorithm 3.1 Process-behaviour chart algorithm.

The process-behaviour chart algorithm (Chandesa et al., 2009a, Chandesa et al., 2009b) has been applied to the test videos described in Chapter 2. For each test video, the Shewhart chart monitors the fluctuating behaviour in the clustered particle deviation and clustered particle speed. Results are shown below.

3.5.2. Normal tracking

Shewhart control chart results when tracking: a clearly visible yellow circle are shown in Figure 3.45 and Figure 3.46, a yellow circle amid clutter are shown in Figure 3.47 and Figure 3.48 and the tennis ball moving normally are shown in Figure 3.49 and Figure 3.50. Figure 3.45, Figure 3.47 and Figure 3.49 monitor fluctuating behaviour in the clustered particle deviation, while Figure 3.46, Figure 3.48 and Figure 3.50 monitor fluctuating behaviour in the clustered particle speed.

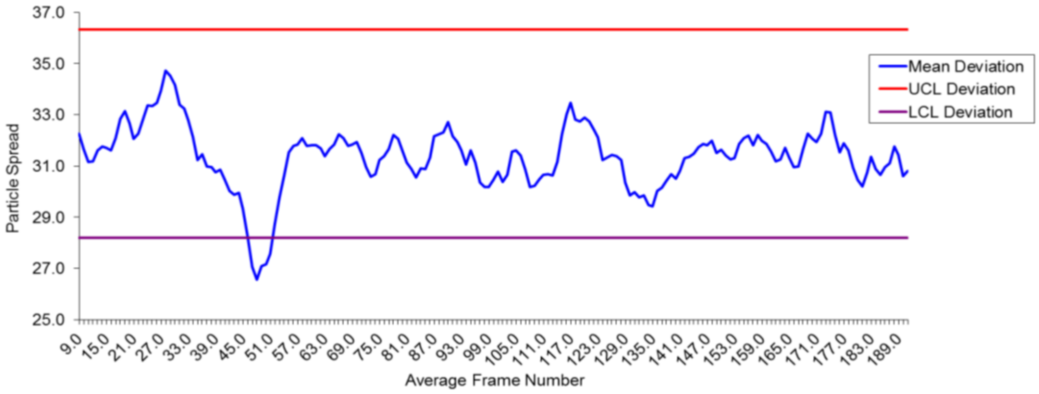


Figure 3.45 Shewhart control chart for particle cluster deviation of a clearly visible yellow circle.

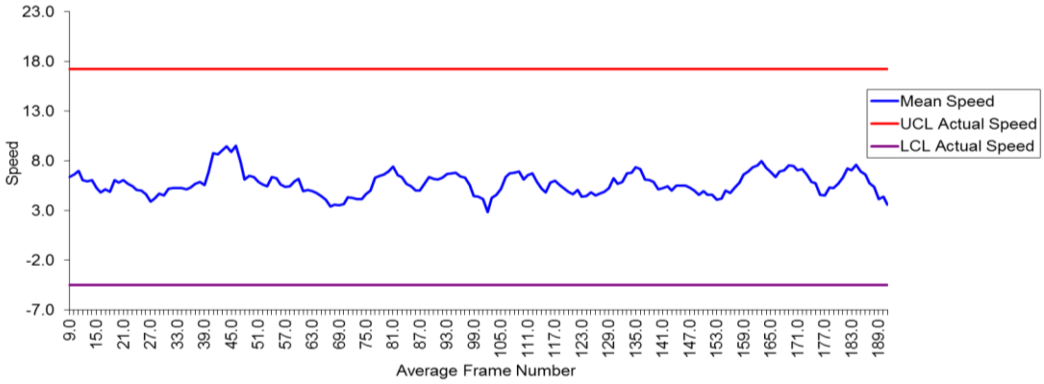


Figure 3.46 Shewhart control chart for particle cluster speed of a clearly visible yellow circle**.**

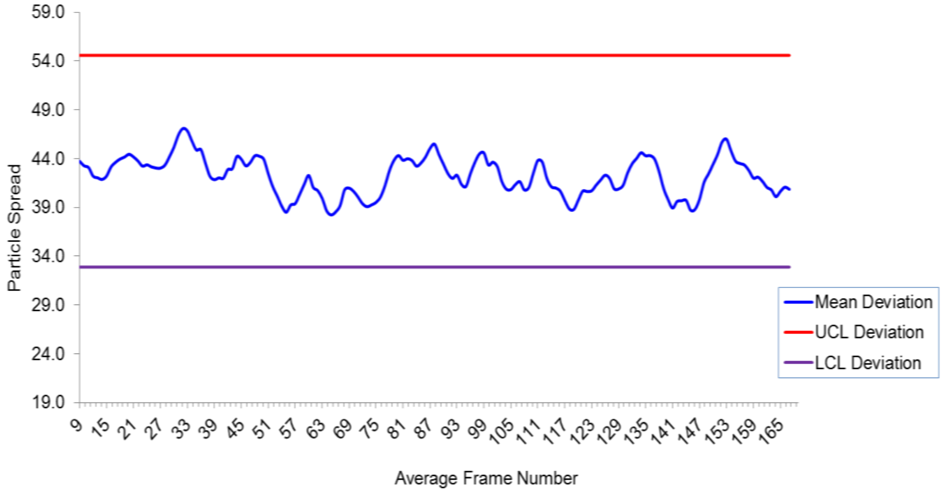


Figure 3.47 Shewhart control chart for particle cluster deviation of a yellow circle moving amid clutter.

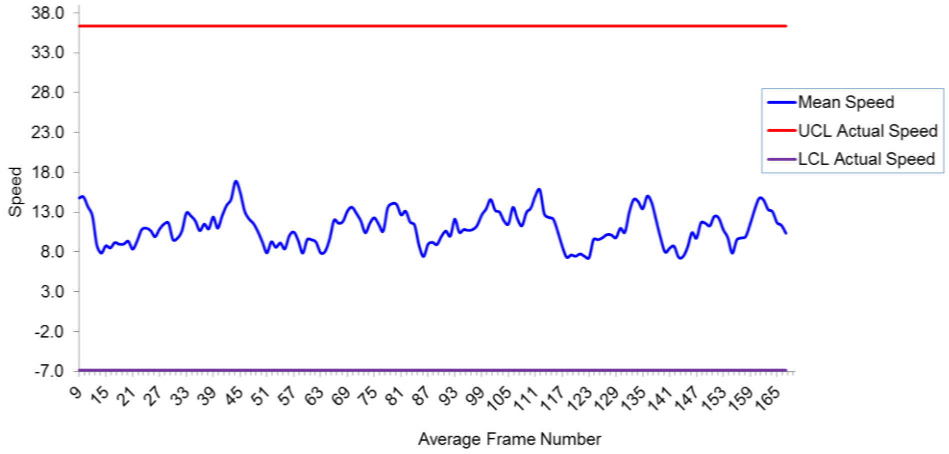


Figure 3.48 Shewhart control chart for particle cluster speed of a yellow circle moving amid clutter.

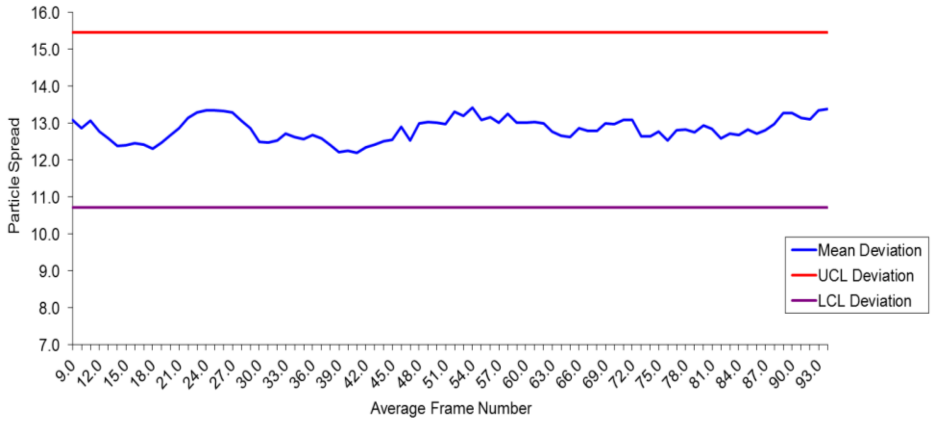


Figure 3.49 Shewhart control chart for particle cluster deviation of a clearly visible tennis ball.

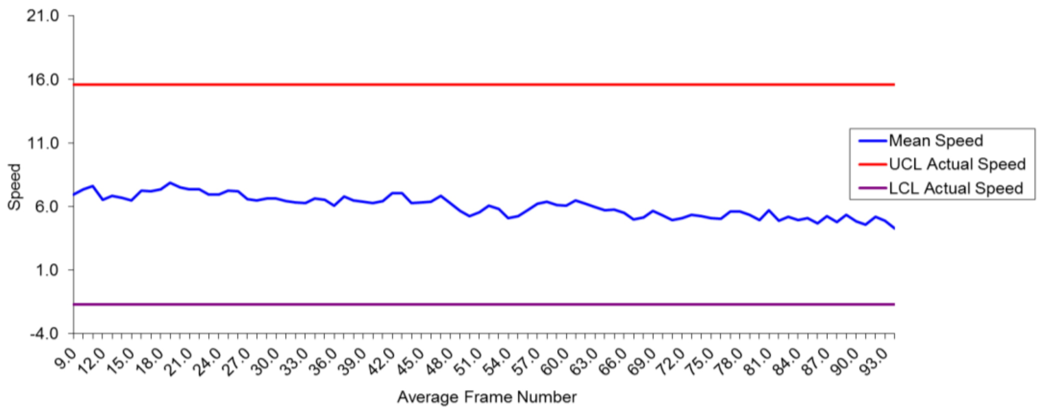


Figure 3.50 Shewhart control chart for particle cluster speed of a clearly visible tennis ball.

Analysis of all the graphs shows that when the target(s) is clearly visible and moving normally:

- The cluster deviation remains within the control limits of the Shewhart control chart.
- The cluster speed also stays within the boundaries of the Shewhart control chart limits.
- Though there is more variation in the data in the presence of clutter, the method appears not to report false positive.

3.5.3. Occlusion

Shewhart control chart results for tracking: a yellow circle as occlusion occurs are shown in Figure 3.51 and Figure 3.52, tracking a woman as she becomes occluded by a tree are shown in Figure 3.53 and Figure 3.54, tracking the tennis ball as occlusion occurs are shown in Figure 3.55 and Figure 3.56 and tracking a table tennis bat as it experiences self-occlusion are shown in Figure 3.57 and Figure 3.58. Figure 3.51, Figure 3.53, Figure 3.55 and Figure 3.57 monitor the fluctuation behaviour in the particle cluster deviation, while Figure 3.52, Figure 3.54, Figure 3.56 and Figure 3.58 monitor the fluctuation behaviour in the particle cluster speed.

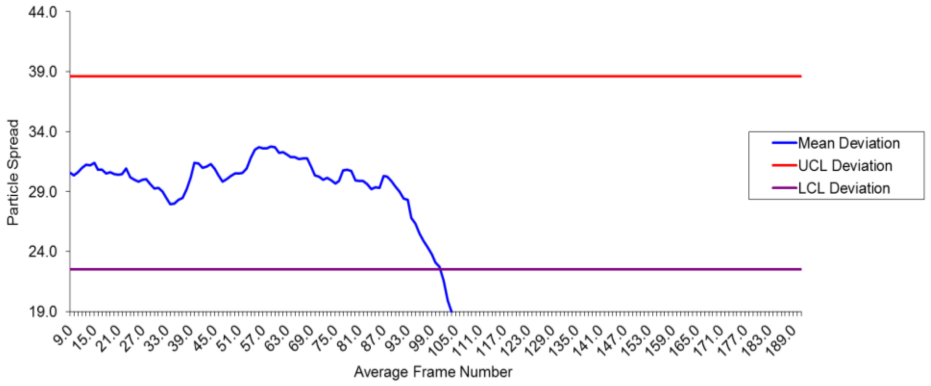


Figure 3.51 Shewhart control chart for particle cluster deviation when tracking a yellow circle as occlusion occurs.

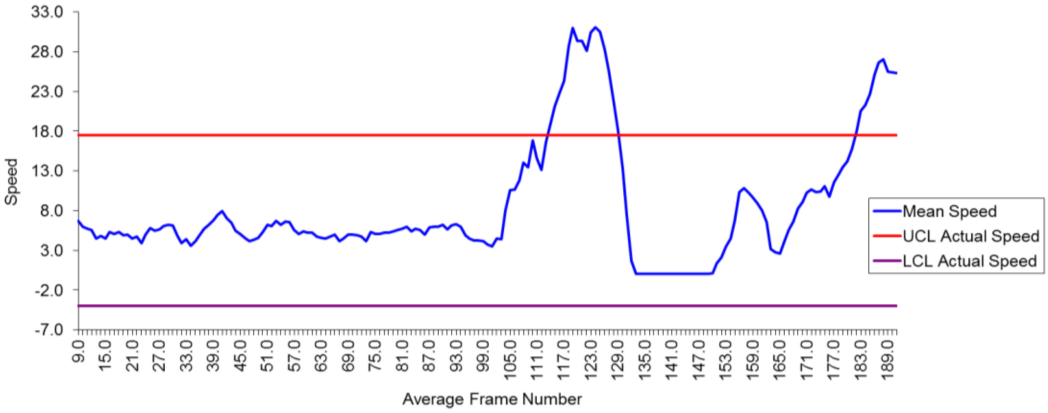


Figure 3.52 Shewhart control chart for particle cluster speed when tracking a yellow circle as occlusion occurs.

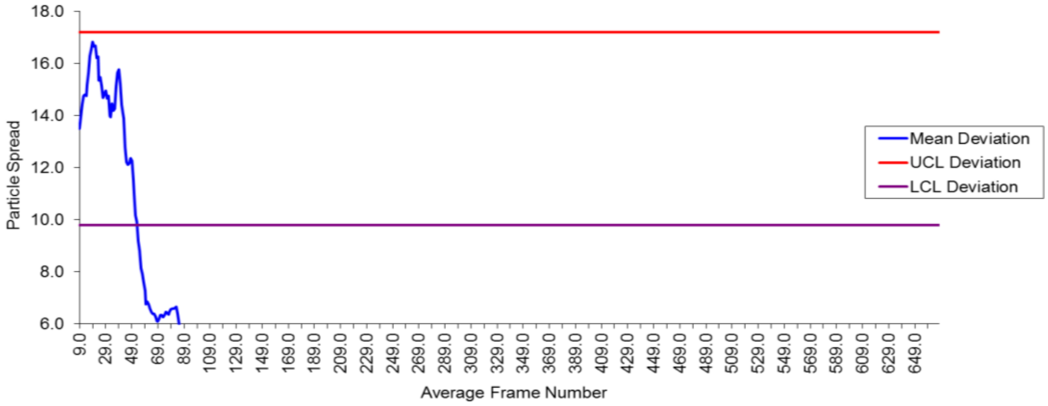


Figure 3.53 Shewhart control chart for particle cluster deviation when tracking a woman as she becomes occluded by a tree.

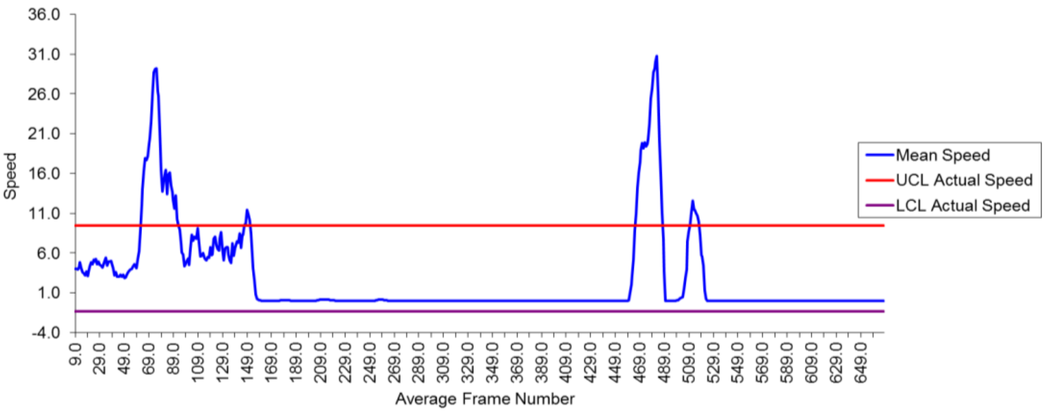


Figure 3.54 Shewhart control chart for particle cluster speed when tracking a woman as she becomes occluded by a tree**.**

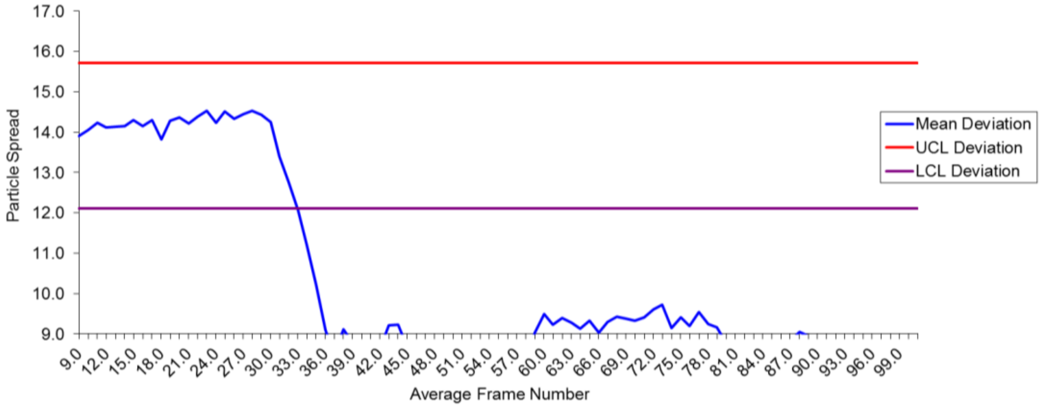


Figure 3.55 Shewhart control chart for particle cluster deviation when tracking a tennis ball as occlusion occurs.

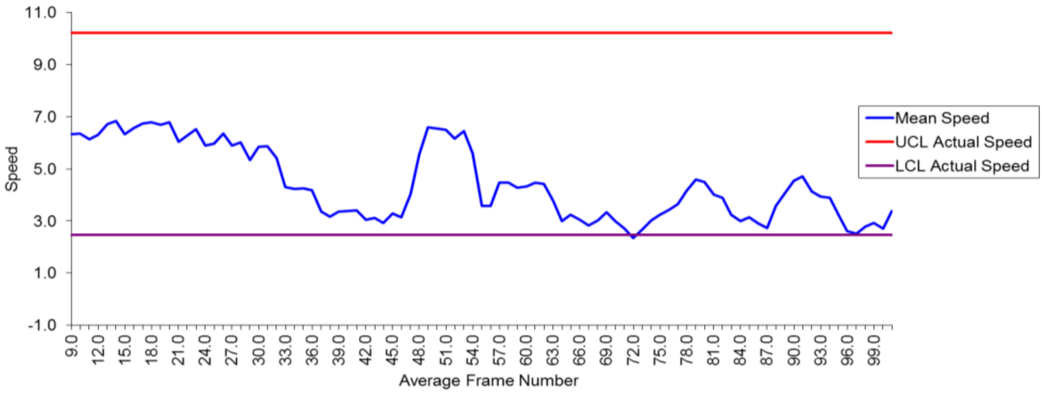


Figure 3.56 Shewhart control chart for particle cluster speed when tracking a tennis ball as occlusion occurs.

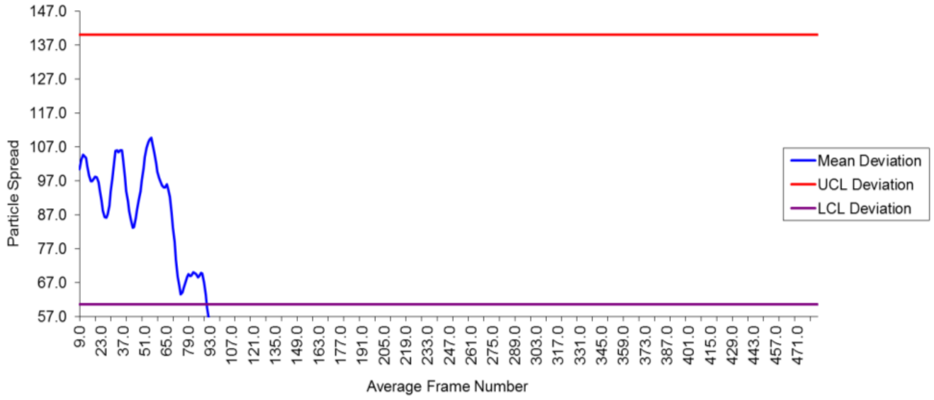


Figure 3.57 Shewhart control chart for particle cluster deviation when tracking a table tennis bat as it experiences self-occlusion.

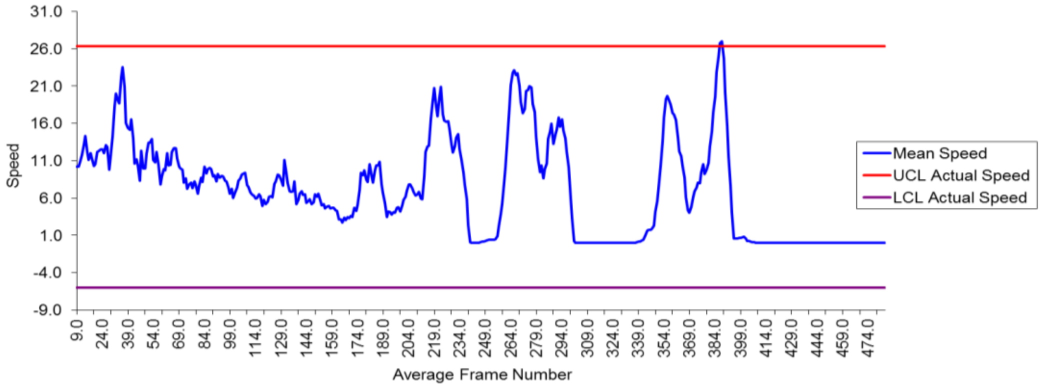


Figure 3.58 Shewhart control chart for particle cluster speed when tracking a table tennis bat as it experiences self-occlusion.

When a target (yellow circle, woman, tennis ball and bat) experiences occlusion, the graphs show that:

- The particle cluster deviation gradually decreases and eventually exceeds the lower control limits of the Shewhart chart.
- The cluster speed in contrast increases and exceeds the upper control limits of the Shewhart chart in graphs generated for tracking a yellow circle (Figure 3.52) and woman (Figure 3.54). Though, the same effect is not produced when tracking the tennis ball (Figure 3.56) and bat (Figure 3.58). In these videos, the graphs show the cluster mean well-correlated and commonly stays within the control limit.

3.5.4. Camouflage

Shewhart control chart results for tracking: a yellow circle as camouflage occurs is shown in Figure 3.59 and Figure 3.60, a football becoming camouflaged by a player is shown in Figure 3.61 and Figure 3.62 and a tennis ball as camouflage occurs is shown in Figure 3.63 and Figure 3.64. Figure 3.59, Figure 3.61 and

Figure 3.63 monitor the fluctuation behaviour of the particle cluster deviation, while Figure 3.60, Figure 3.62 and Figure 3.64 monitor the fluctuation behaviour of the particle cluster speed.

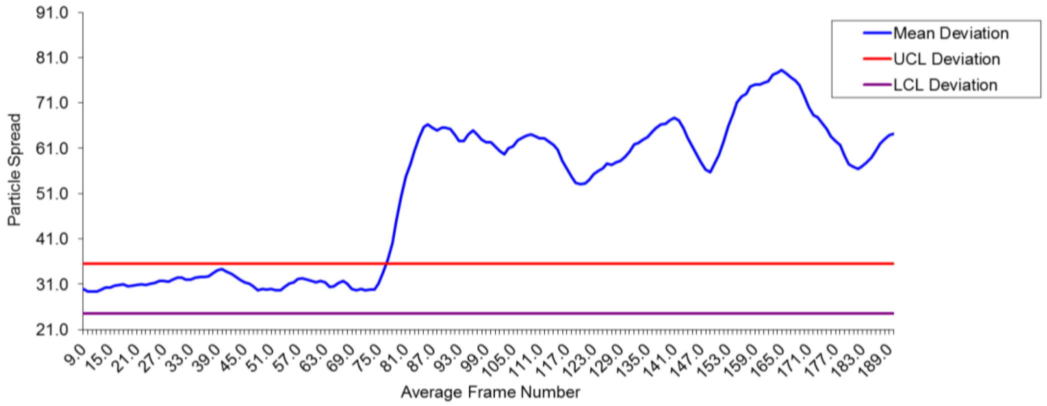


Figure 3.59 Shewhart control chart for particle cluster deviation when tracking a yellow circle as camouflage occurs.

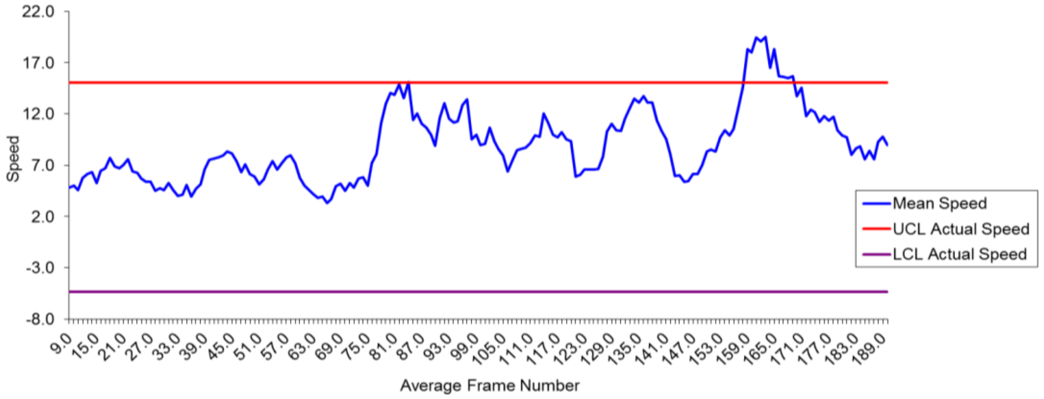


Figure 3.60 Shewhart control chart for particle cluster speed when tracking a yellow circle as camouflage occurs.

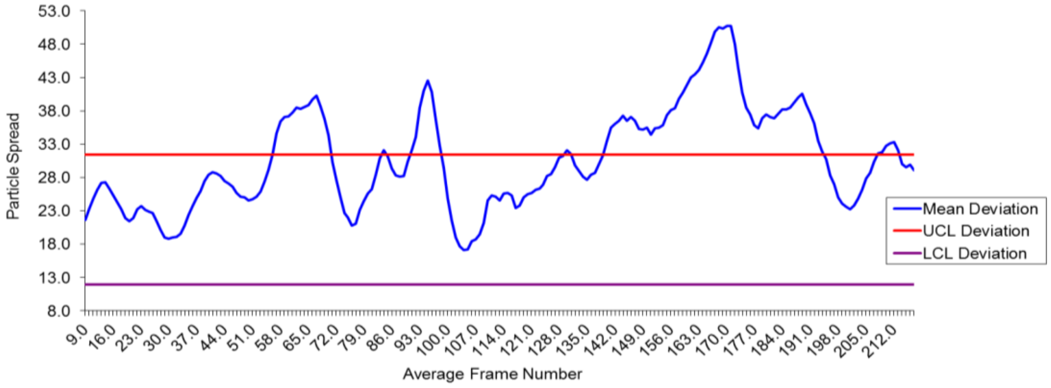


Figure 3.61 Shewhart control chart for clustered particle deviation when tracking a football becomes camouflaged by a player.

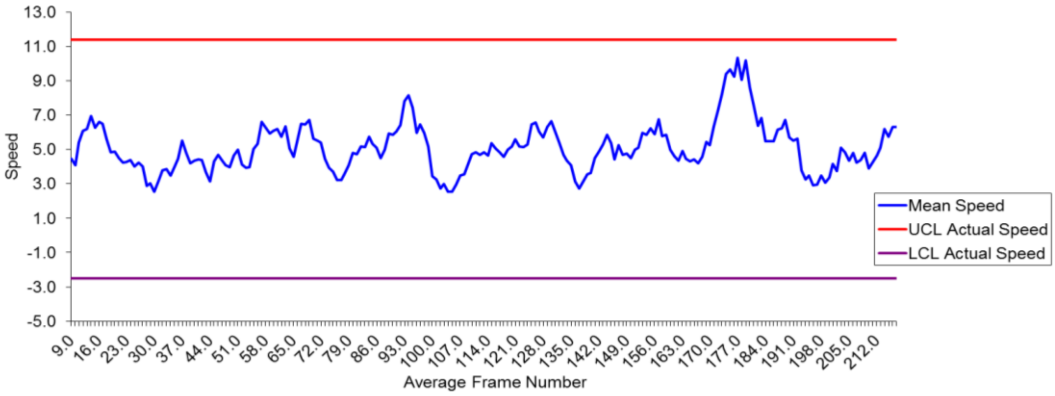


Figure 3.62 Shewhart control chart for particle cluster speed when tracking a football that becomes camouflaged by a player.

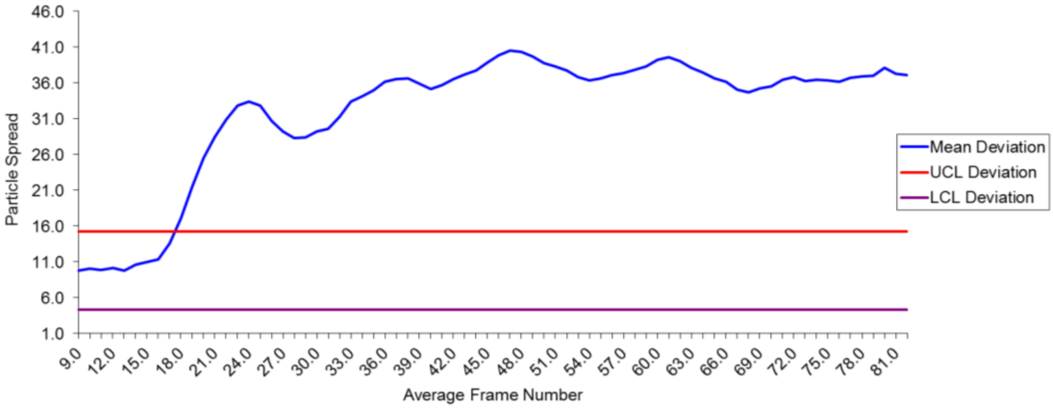


Figure 3.63 Shewhart control chart for particle cluster deviation when tracking a tennis ball as camouflage occurs.

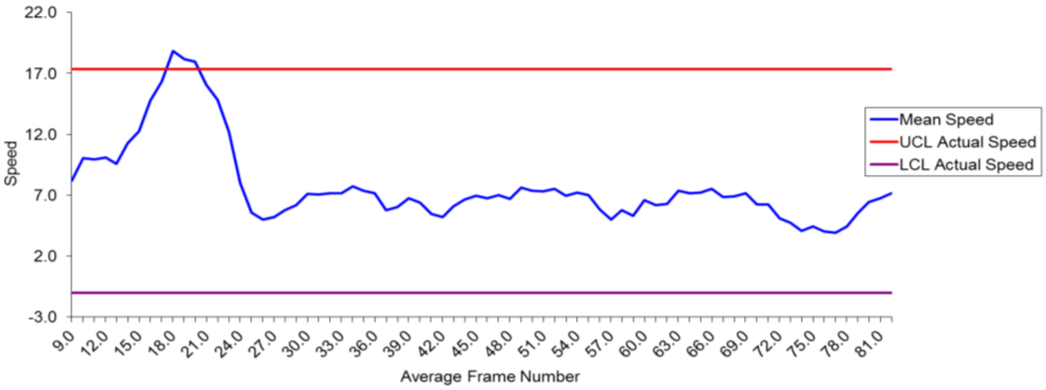


Figure 3.64 Shewhart control chart for particle cluster speed when tracking a tennis ball as camouflage occurs.

Graphs generated when a target (yellow circle, football and tennis ball) experiences camouflage show that:

- The particle cluster deviation progressively increases and exceeds the upper Shewhart control chart limits.

- There is, however, variation in the results generated for monitoring the particle cluster speed. In video showing a yellow circle (Figure 3.60) and tennis ball (Figure 3.64), the cluster speed also increases and exceeds the upper control limits. But the same effect is not recorded for video showing the camouflaged football (Figure 3.62). Here, the cluster speed remains well-correlated and stays within the control limits.

3.5.5. Summary

Analysis shows that particle cluster deviation provides a clearer indicator of the occurrence of occlusion and/or camouflage than do apparent changes in speed. Though speed may go out of bounds during an occlusion or camouflage event, this cannot be relied upon to happen in even simple cases, and is at best a secondary effect. As a result, particle cluster deviations are used as the primary tool to determine when occlusion and camouflage occurs, with speed data providing supplementary information.

Analysis also shows that by using the process-behaviour chart, the information gathered can be exploited to allow the tracker to determine when a target is experiencing occlusion and/or camouflage by monitoring the control points.

3.6. Evaluation

Artificial videos with clutter present were created to evaluate the robustness of the process-behaviour chart tracker (also known as control chart tracker). The contents of these artificial videos are identical to the contents of the artificial videos with clutter present discussed in section 2.4.2.1 of Chapter 2. Three different sets were created exhibiting a target either moving normally, or experiencing occlusion or experiencing camouflage. In each set, a total of six artificial videos were created for each clutter type: background clutter, image clutter, motion clutter and target clutter. Discussion into the characteristics of each clutter type has already been presented in section 2.4.2 of Chapter 2.

For each clutter type, the amount of clutter generated in each of the six artificial videos is controlled by a percentage. The percentage level increases progressively from a minimum percentage value to a maximum percentage value. The different percentage levels are evenly spaced in between. A minimum percentage value implies the lowest value the tracking system can handle without failing. A maximum percentage value implies: (1) the value is the highest value accessible which is 100% or (2) the value is the limit before the tracking system fails. Trial and error using percentage values ranging from 0.1% to 100% was performed to determine the minimum and maximum percentage values as well as the different percentage levels in between. The values are:

- Background clutter: 10%, 30%, 50%, 70%, 90% and 100%.
- Image clutter: 10%, 30%, 50%, 70%, 90% and 100%.
- Motion clutter: 1.0%, 2.5%, 5.0%, 7.5%, 9.0% and 10.0%.
- Target clutter: 0.14%, 0.28%, 0.56%, 0.98%, 1.12%, and 1.26%.

These percentage levels are represented as diamond shape patterns in the graphs. Figures illustrating the above mentioned percentage levels used when generating the six artificial videos for each clutter type can be found in Appendix C of this thesis.

The robustness of the control chart tracker is determined by running the tracker on each artificial video five times and at each time, recording the following results:

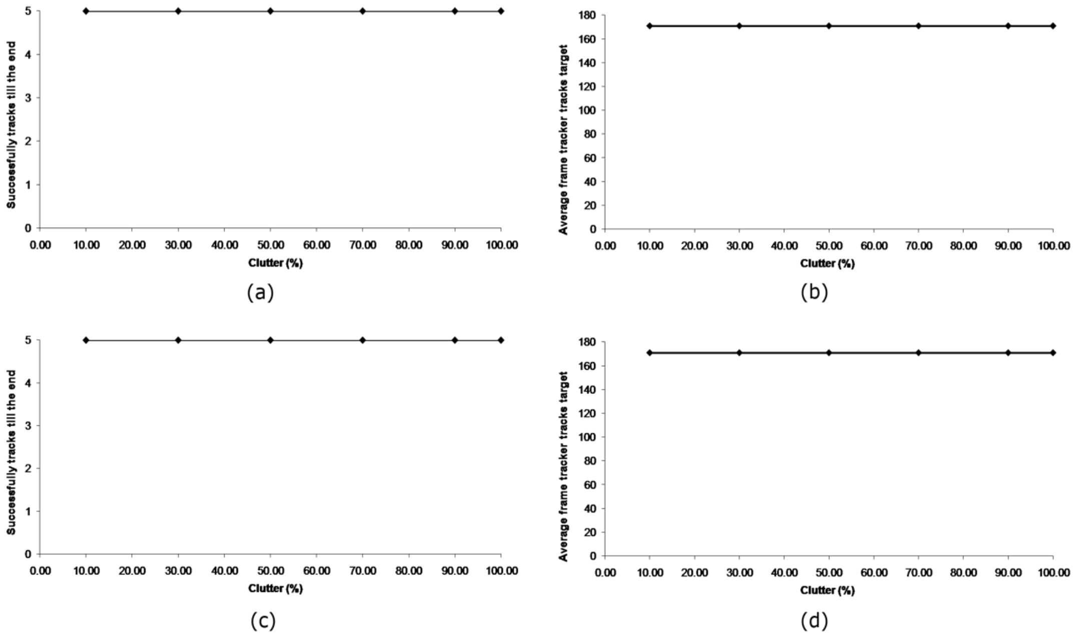
1) The success rate at successfully: (1) tracking a target moving normally for the entire image sequence, detecting the occurrence of (2) occlusion and (3) camouflage.
2) The mean and standard deviation of the number of frames for which tracking is achieved. High mean values are sought when tracking normally (indicating longer periods of successful tracking) and low deviations when occlusion and camouflage are present (indicating consistent detection of these events). The results are discussed subsequently.

The control chart tracker is a point based implementation. Thus, the initial starting point for each artificial test video is user defined. For every artificial test videos, a particle set size of 100 is used. Since the target to be tracked in every artificial test videos is a yellow circle, hence, using a radius size of 3 is found to produce good tracking result. A histogram bin size of 10 is selected for all artificial videos.

3.6.1. Normal tracking

In this analysis, the Condensation tracker was also applied onto the above mentioned artificial videos with clutter present. The robustness of the Condensation tracker is evaluated in the same manner as the control chart tracker. The Condensation tracker is also a point based implementation and utilised the same parameters settings used by the control chart tracker.

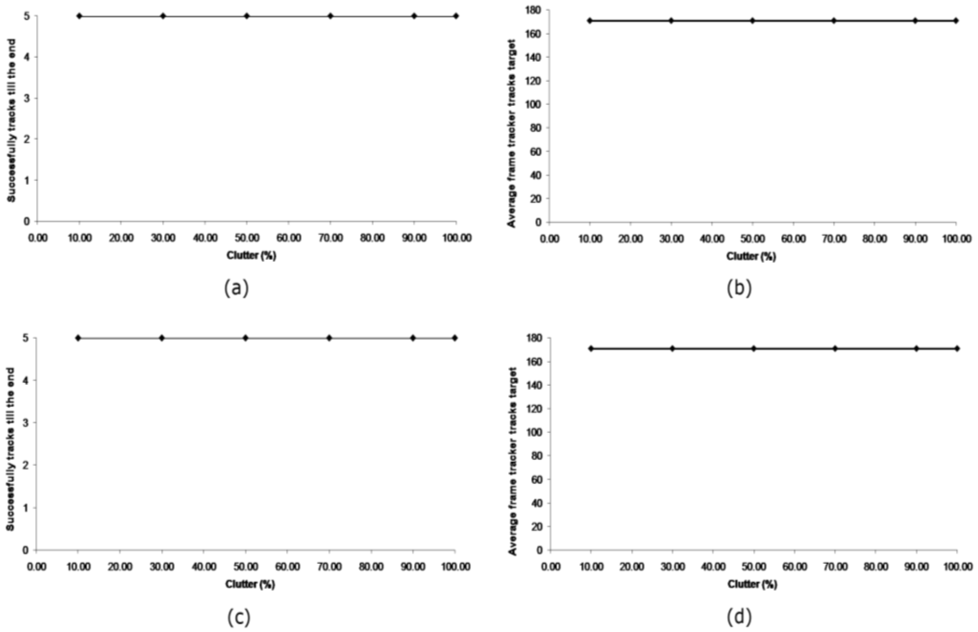
Figure 3.65, Figure 3.66, Figure 3.67, and Figure 3.68 present graphs showing the robustness of the Condensation tracker and the control chart tracker at tracking target(s) moving normally amid the presence of different types of clutter.



(a) (b)

(c) (d)

Figure 3.65 Graphs showing: (1) normal tracking success rate using: (a) Condensation tracker; (c) control chart tracker and (2) mean number of frames tracking a target correctly using: (b) Condensation tracker; (d) control chart tracker; while normal tracking amid the presence of background clutter.

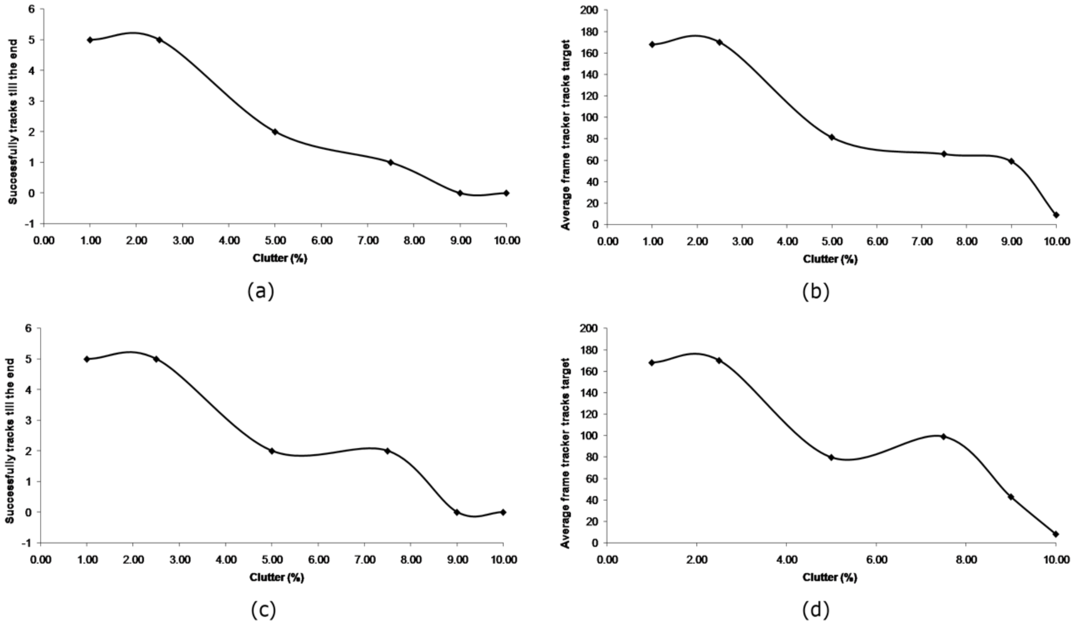


(a) (b)

(c) (d)

Figure 3.66 Graphs showing: (1) normal tracking success rate using: (a) Condensation tracker; (c) control chart tracker and (2) mean number of frames

tracking a target correctly using: (b) Condensation tracker; (d) control chart tracker; while normal tracking amid the presence of image clutter.



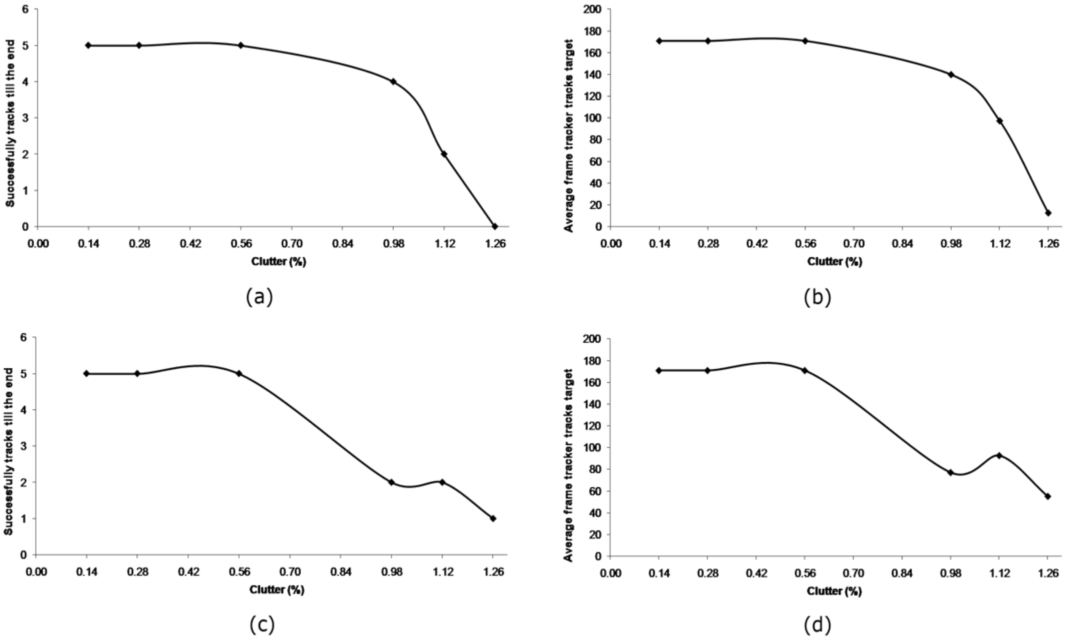
(a)

(b)

(c)

(d)

Figure 3.67 Graphs showing: (1) normal tracking success rate using: (a) Condensation tracker; (c) control chart tracker and (2) mean number of frames tracking a target correctly using: (b) Condensation tracker; (d) control chart tracker; while normal tracking amid the presence of movement clutter.



(a)

(b)

(c)

(d)

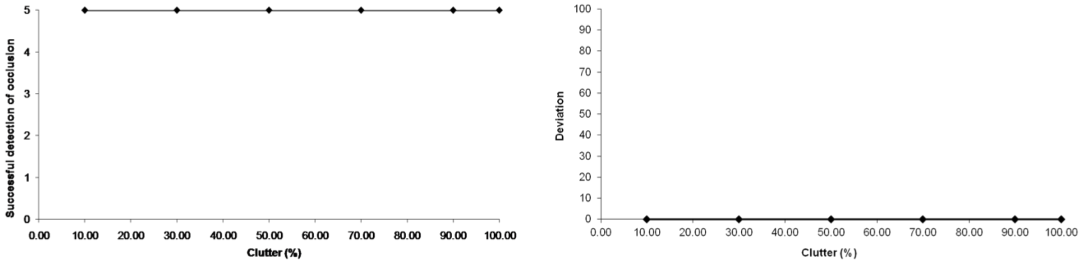
Figure 3.68 Graphs showing: (1) normal tracking success rate using: (a) Condensation tracker; (c) control chart tracker and (2) mean number of frames tracking a target correctly using: (b) Condensation tracker; (d) control chart tracker; while normal tracking amid the presence of target clutter.

Analysis shows that:

- The control chart tracker is equally robust as the Condensation tracker when used to track a target moving normally amid the presence of different percentage of background (Figure 3.65) and image (Figure 3.66) clutter. Both trackers effectively track the target throughout the maximum frames within an image sequence in all 5 attempts.
- However, there seem to be variations in regards to the robustness of the control chart tracker as well as the Condensation tracker when tracking a target moving normally amid the presence of motion and target clutter. In Figure 3.67 and Figure 3.68, the graph shows that both trackers suffer a gradual decrease in: (1) continuing to keep track of the target; and (2) the average number of frames a target is tracked correctly, for the duration of the entire image sequence. When the percentage of motion clutter and target clutter is increased, both trackers tend to lose the target early on and never track the target for the entire image sequence. This contributes to the gradual decrease recorded for the graphs in Figure 3.67 and Figure 3.68.

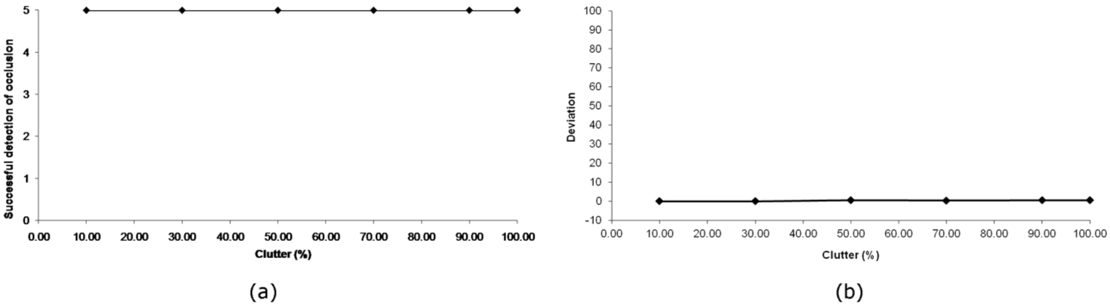
3.6.2. Occlusion

Graphs showing the robustness of the control chart tracker at tracking target(s) amid the presence of an occluding object and different types of clutter are presented in Figure 3.69, Figure 3.70, Figure 3.71 and Figure 3.72.



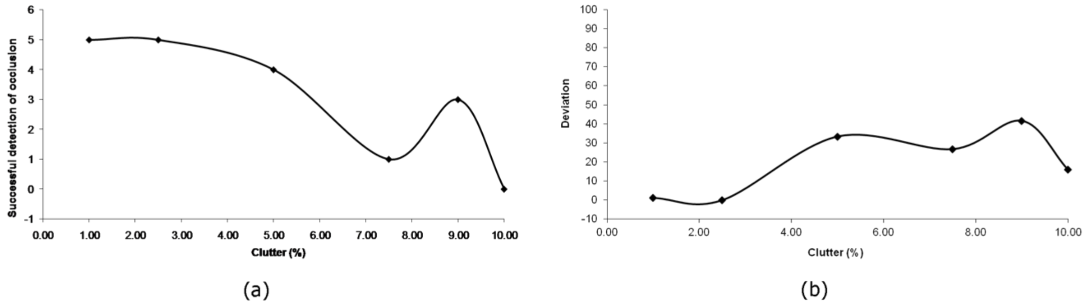
(a) (b)

Figure 3.69 Graphs showing: (a) the success rate of detecting occlusion and (b) the standard deviation of the number of frames at which occlusion is detected; while in the presence of an occluding object and background clutter.



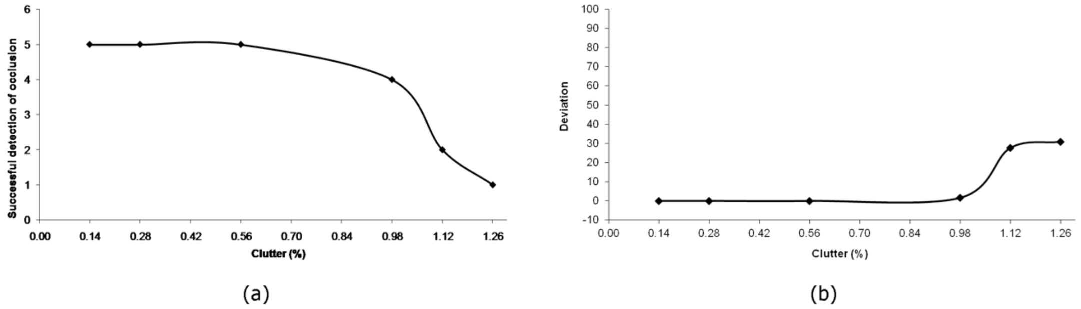
(a) (b)

Figure 3.70 Graphs showing: (a) the success rate of detecting occlusion and (b) the standard deviation of the number of frames at which occlusion is detected; while in the presence of an occluding object and image clutter.



(a) (b)

Figure 3.71 Graphs showing: (a) the success rate of detecting occlusion and (b) the standard deviation of the number of frames at which occlusion is detected; while in the presence of an occluding object and movement clutter.



(a) (b)

Figure 3.72 Graphs showing: (a) the success rate of detecting occlusion and (b) the standard deviation of the number of frames at which occlusion is detected; while in the presence of an occluding object and target clutter.

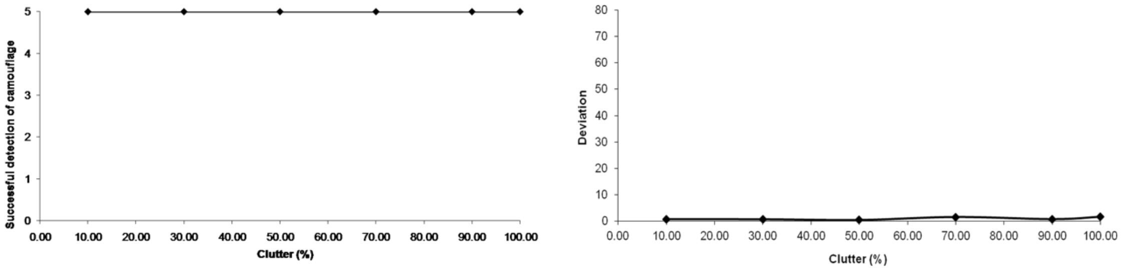
Analysing each graph shows that:

- The control chart tracker is equally robust at detecting occlusion amid the presence of different percentage of background (Figure 3.69) and image (Figure 3.70) clutter. The control chart tracker successfully detects occlusion in all 5 attempts. Consideration of the deviation in the frame number at which events are reported shows a low deviation which is consistent with accurate detection of occlusion.

- However, a systematic drop in the robustness of the control chart tracker is recorded when the percentage of movement (Figure 3.71) and target (Figure 3.72) clutter is increased. As the clutter percentage is increased, the control chart tracker starts to lose the target early on, far ahead of the actual occurrence of occlusion. This eventually leads to the control chart tracker not successfully detecting occlusion in every attempt. Consideration of the deviation shows an increase in deviation being recorded as the detector becomes less reliable at detecting occlusion as a result of clutter.

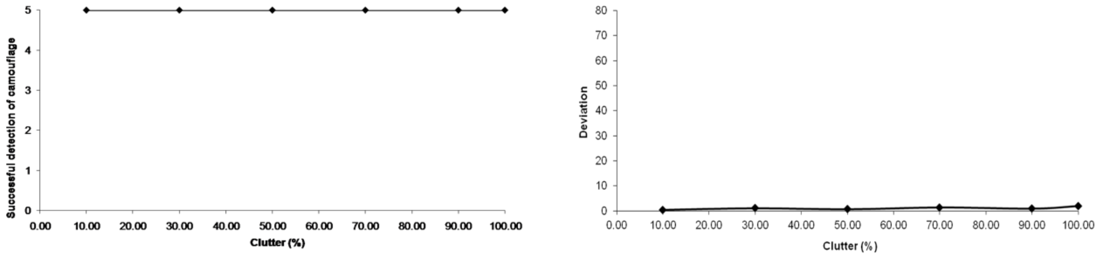
3.6.3. Camouflage

Figure 3.73, Figure 3.74, Figure 3.75 and Figure 3.76 present graphs showing the robustness of the control chart tracker at tracking target(s) amid the presence of a camouflaging object and different types of clutter.



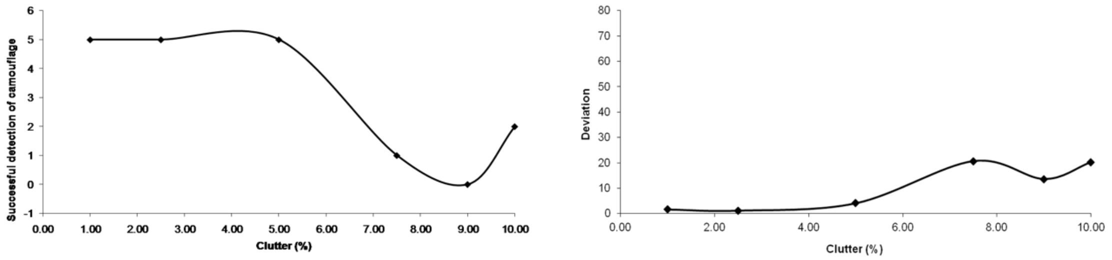
(a) (b)

Figure 3.73 Graphs showing: (a) the success rate of detecting camouflage and (b) the standard deviation of the number of frames at which camouflage is detected; while in the presence of a camouflaging object and background clutter.



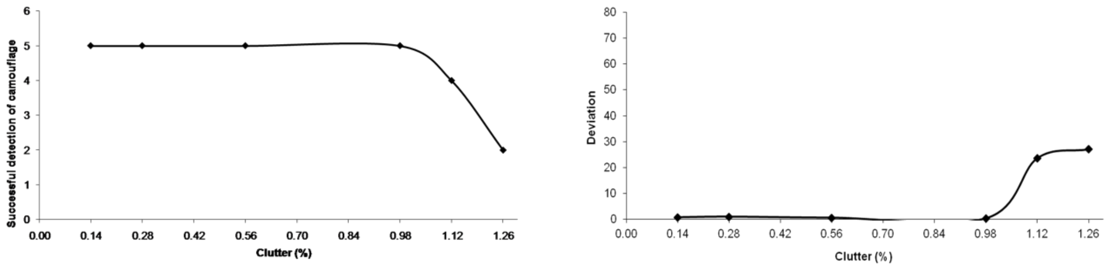
(a) (b)

Figure 3.74 Graphs showing: (a) the success rate of detecting camouflage and (b) the standard deviation of the number of frames at which camouflage is detected; while in the presence of a camouflaging object and image clutter.



(a) (b)

Figure 3.75 Graphs showing: (a) the success rate of detecting camouflage and (b) the standard deviation of the number of frames at which camouflage is detected; while in the presence of a camouflaging object and movement clutter.



(a) (b)

Figure 3.76 Graphs showing: (a) the success rate of detecting camouflage and (b) the standard deviation of the number of frames at which camouflage is detected; while in the presence of a camouflaging object and target clutter.

Analysis provides evidence that:

- The control chart tracker is equally robust at detecting camouflage amid the presence of different percentage of background (Figure 3.73) and image (Figure 3.74) clutter. The control chart tracker successfully detects camouflage in all 5 attempts. Consideration of the associated deviation shows a low deviation is recorded when detecting camouflage. The trivial variation in the recorded deviation is due to the tracker not detecting camouflage at the same frame number in all 5 attempts.
- The control chart tracker seems to be less robust at detecting camouflage amid the presence of different percentage of motion (Figure 3.75) and target (Figure 3.76) clutter. The success rate of the control chart tracker detecting camouflage is shown to decrease as clutter percentage is gradually increased. The gradual decrease in robustness is influenced by the tracker failing to detect camouflage successfully in every attempt, as a result of the tracker losing the targets early on, due to the increased amount of clutter used. Consequently, this contributes to a high deviation being recorded as the detector becomes less reliable and not detect the event of camouflage.

3.6.4. Summary

The evaluation analysis provides evidence that:

- In normal tracking, the control chart tracker is shown to be equally robust as the Condensation tracker at tracking a target for the entire image sequence, while in the presence of background and image clutter. Both trackers also showed to be effective at tracking the target for the maximum number of frames within an image sequence amid the presence of clutter. However, both trackers are shown to be equally less robust when tracking targets amid the presence of movement and target clutter. Analysis showed that as the percentage clutter values are gradually increased, the robustness of both trackers gradually decreases. Both trackers tend to lose the target early on, which results in both trackers not continuously tracking the target for the entire image sequence and are also less effective at tracking the target for the maximum number of frames within the image sequence at every attempt.
- In detecting occlusion and camouflage, the control chart tracker is shown to be robust at detecting these events amid the presence of background and image clutter. In all five attempts, the control chart tracker successfully detects occlusion and camouflage. This also contributes to a low deviation being recorded, which is consistent with the detection of these events. However, analysis also showed the control chart tracker to be less robust at detecting occlusion or camouflage while in the presence of movement and target clutter. Analysis showed that as the percentage clutter values are gradually increased, the robustness of the tracker gradually decreases. This is due to the control chart tracker losing track of the target early on as a result of clutter and does not successfully detect occlusion or camouflage. Due to the tracker losing the target too early, also contributes to a high deviation being recorded, which indicates that the tracker has become less reliable at detecting these events.

In summary, this analysis provides evidence that in spite of the excessive amount of clutter (movement and target clutter) added; the process-behaviour chart tracker is as robust as the Condensation when the target is fully visible. Moreover, results presented in sections 3.6.2 and 3.6.3 showed the process-behaviour chart tracker to have the added ability to detect occlusion and camouflage amid the presence of different clutter types.

3.7. Chapter summary

Literature has shown that building a Gaussian Mixture Model using EM algorithm to cluster particle set is not the norm in visual tracking. Analysis provide evidences that examining particles collectively during an image sequence provides vital information pertaining to the particle set behaviour while tracking a target moving normally or experiencing occlusion and/or camouflage. Analysis also proved that in spite of using either a two-dimension EM algorithm or a three-dimension EM algorithm, both approaches presented similar outcome in identifying cues related to these events. Analysis then provided attestation that by using an approach called, process-behaviour chart, the information gathered by the EM algorithm at every time-step of an image sequence can be accessed to allow a tracker to determine when a target is being tracked moving normally or experiencing occlusion and/or camouflage. Further analysis also showed that the process-behaviour chart tracker to be as robust as the Condensation tracker in normal tracking. Also, process-behaviour chart tracker is shown to have the added ability at detecting occlusion and camouflage amid the presence of different types of clutter used.

In Chapter 5, the approach presented in this chapter is applied to detect the occurrences of occlusion and/or camouflage when tracking multiple targets moving within a static scene.

Chapter 4 Measuring Particle Spread

4.1. Aim

The aims of this chapter are to:

1. Present the notion of the Particle History Image (PHI), a view-based representation of apparent motion constructed from the particle sets employed by a particle filter-based tracking algorithm.
2. Apply a texture edge detection algorithm to the PHI, presenting the notion of the Particle Boundary Image (PBI).
3. Examine the PHI and PBI to provide information on the width of the particle spread. Two ways to analyse these data structures are considered: temporal and spatial.
4. Combine PHI and PBI-based measurements with a process-behaviour chart to detect occlusion and camouflage during tracking.

4.2. Motivation

Motion analysis and action recognition require compact but rich representations of the spatial configuration and movement of the object(s) of interest over some time period. One such representation, which has received considerable attention in the literature, is the temporal template of (Bobick et al., 1997, Davis et al., 1997, Bobick et al., 1996b, Bobick et al., 2001).

Temporal templates provide a view-based representation intended to support direct recognition of the two-dimension patterns of motion associated with known three-dimension actions and events. The approach is motivated by the observation that humans can easily recognise actions given only very low resolution image sequences that provide no information about the three-dimension structure of the viewed scene (Bobick et al., 1996a). Temporal templates have two components: a Motion Energy Image (MEI) and a Motion History Image (MHI).

Motion Energy Images are image-based representations showing the cumulative motion at each pixel location over some time period. Originally, these were obtained by computing the sum of squared difference between the first and each subsequent image, applying a threshold and summing the resulting binary images. The MEI then reflects the number of frames in which a given pixel showed above threshold motion. (Bobick et al., 1996a) use MEIs to identify actions likely to be taking place, before a second, verification stage completes recognition. Later work in (Davis et al., 1997) lead to the identification of the binary MEI, in which the union, not the sum of the constituent binary images is recorded. Rather than note where motion is taking place, Motion History Images (Davis et al., 1997) record recency of motion, capturing how movement progresses. If $D(x,y,t)$ is a binary image sequence indicating region(s) of motion over $0 \le t \le T-1$, the binary Motion Energy Image $E_T(x,y,t)$ is:

$$E_T(x,y,t) = \bigcup_{i=0}^{T-1} D(x,y,t-i) \qquad (4.1)$$

While the corresponding Motion History Image $H_T(x,y,t)$ is:

$$H_T(x,y,t) = \begin{cases} T \text{ if } D(x,y,t) = 1\text{, i.e. if motion is detected at time } t \\ \max\left(0, H_T(x,y,t-1)-1\right)\text{, otherwise} \end{cases}$$

Note that the binary MEI is obtained by thresholding the MHI above 0. This simple relationship has marked the MHI out as the key representation, and lead to it receiving more attention in subsequent literature.

A number of extensions to/variations on the Motion History Image theme have been proposed. (Babu et al., 2004) construct MHIs from MPEG compressed video data, partially decoding the MPEG to extract vectors which are used to construct coarse MHIs, at the macroblock level. Motion flow histories are also constructed which record the horizontal and vertical components of motion at time t as long as the value reported is within the threshold of local, mean filtered measurements. MHIs and motion flow histories are used together to underpin recognition of simple human actions such as bending and walking. (Bradski et al., 2000, Bradski et al., 2002) present timed MHIs, in which pixel values record actual time, rather than frame number. (Davis, 1999) present an approach that provides better patterns of motions description when recognizing various human movements by means of characterising a MHI through multiple, overlapping histograms of motion orientations. (Davis, 2001) defines hierarchical Motion History Images, building a MHI pyramid before extracting local motion histograms with which to support recognition. (Ahad et al., 2007) extend the MHI by considering the direction, and not just presence, of motion during its construction. They estimate motion in each of four directions, producing separate MHIs for each and arguing that the representation obtained is less sensitive to self-occlusion and so leads to better activity recognition on more complex scenes.

(Weinland et al., 2006) have extended the concept to three dimensions, basing their representation on the visual hulls of objects viewed by multiple cameras.

Motion History Images have found application in a variety of situations. (Davis et al., 1997, Davis et al., 1998) recognise aerobic exercises, and deployed MEIs in an early version of their KidsRoom interactive environment (Bobick et al., 1999). (Bradski et al., 2000, Bradski et al., 2002) recognise walking, waving and clapping gestures, allowing a freestanding person to conduct a virtual orchestra. (Boehme et al., 1998) use MHIs in the user identification component of their mobile robot system.

The main strength of the Motion History Image approach is that it neatly captures motion information across the whole field of view and over a wide range of time periods. While MHIs created over very short periods perhaps fail to take best advantage of the method, and those describing over very long periods can be hard to interpret (see e.g. motion context (Zhang et al., 2008)), MHIs can succinctly represent motion over the times taken to form many interesting gestures and actions. As (Rosales, 1998) notes: *These representations collapse the temporal component in a way that no explicit temporal or sequence matching is needed* – the use of an image-based representation means that many image processing and analysis tools and methods exist that can be applied to MHI interpretation. (Bradski et al., 2000) use standard image gradient methods to compute local motion direction from their timed MHIs, and pixel dilation and region growing algorithms to perform motion segmentation. Action recognition typically involved extraction of standard image features from (regions of) MHIs. Hu moments (Hu, 1962) were used in Davis and Bobick's original work, and in other subsequent projects (Bobick et al., 1999, Bradski et al., 2000). (Rosales, 1998) assumes moments will be used, and examines the effectiveness of different recognition approaches.

The various form of Motion History Image discussed hitherto, however, share two common limitations:

1. Each is based upon some form of boundary representation of the object(s) of interest. This means that the information they contain is inextricably linked to the shape of the image plane projection of the target object. Examination of the literature suggests that MHIs have only been employed to recognise events involving objects of a single shape class – humans.
2. That boundary representation must be extracted successfully and independently from each frame of the input image sequence as failure to

do so can result in a distorted MHI being produce; the addition of a new silhouette to a developing MHI is a hard decision. Fully reliable background/foreground segmentation is, however, beyond the state of the art unless the environment is constrained in some way.

Section 4.3 presents and investigates the notion of the Particle History Image (PHI), a view-based representation of apparent motion constructed not from silhouettes, but from the particle sets employed by particle filter-based tracking algorithms. The proposed representation are independent of shape and the appearance model used to track the target of interest and, by representing temporal variation in the estimated target state density.

4.3. Particle history image

The fundamental component of a PHI is the particle set. PHIs are generated from the particles' x and y coordinate over a temporal window. If particle set for a time step is defined as $\{P(X_t^n)\}_{n=1}^N$, where $P(X_t^n)$ is the particle representation, *n* is the *n*th particle set in a *N* size population and *t* is the time step, hence, a $PHI(X,T)$ is defined as:

$$PHI(X,T) = \bigcup_{j=t}^{t+T} \{P(X_j^n)\}_{n=1}^N \qquad (4.2)$$

where T refers to the temporal window size and is $0 \leq T \leq M$ where M refers to the maximum time step of an image sequence.

Figure 4.1 shows a PHI captured over a temporal window size of 52 frames for video(s) tracking a target moving normally.

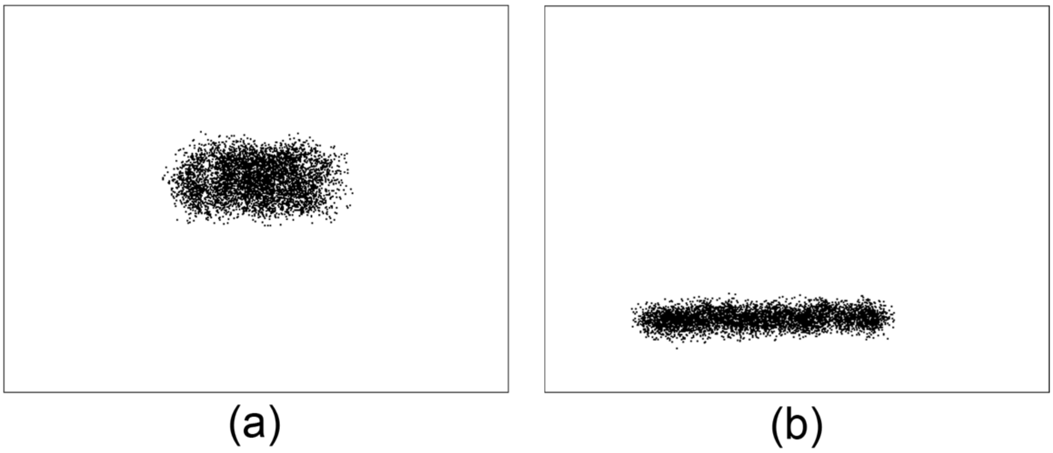


Figure 4.1 PHIs for a: (a) yellow circle moving normally; (b) tennis ball moving normally.

A PHI can be linearly normalised as in (Bradski et al., 2000, Bradski et al., 2002) to illustrate gradual changes in intensity to represent the temporal transition. However, to address the research problem in this thesis, a binary PHI is used.

Generating a PHI over a temporal window exhibiting occlusion or camouflage provides a chronological view-based representation of the change in the particle sets' behaviour as tracking is disrupted. In Figure 4.2, PHIs are produced over particle filtered image sequences exhibiting a yellow circle experiencing occlusion and a tennis ball experiencing camouflage.

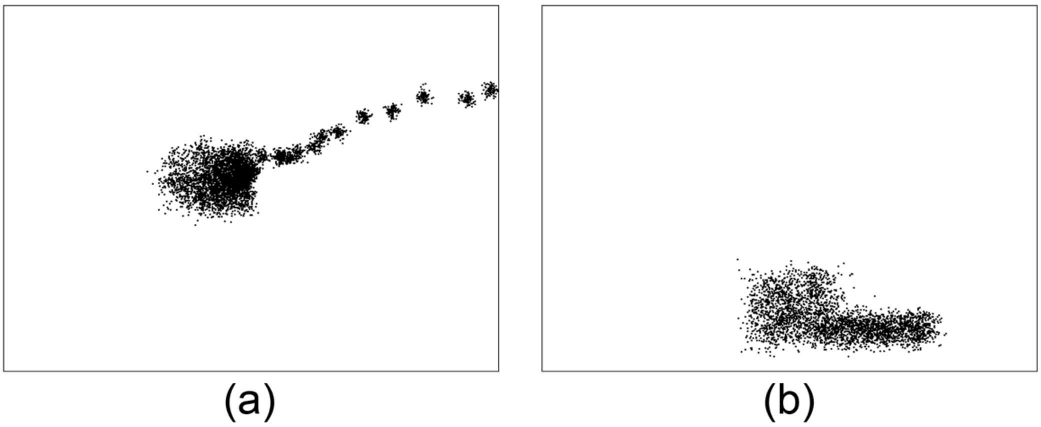


Figure 4.2 PHIs for a: (a) yellow circle experiencing occlusion; (b) tennis ball experiencing camouflage.

Note that as camouflage occurs, the PHIs reflect the camouflaging object's shape as particles begin to spread across it. During occlusion, however, particles tend to closely cluster together and detach themselves from the target when the target becomes fully occluded. Information pertaining to the change in the particle sets's behaviour during disruptive events can be obtained by segmenting out the particle mass in the foreground.

4.4. Particle boundary images

4.4.1. Texture edge detection implementation

A Particle Boundary Image (PBI) is a view-based representation that outlines the boundary surrounding the particle spread in a PHI. A PBI is constructed here by applying a texture edge detection algorithm to the PHI. The texture edge detection algorithm comprises two processes: Laws' texture energy and the Compass operator. Laws' texture energy analyses the texture information within the PHI whereas a Compass operator is used to locate and highlight the texture edges within the PHI. Section 4.4.1.1 and 4.4.1.2 details the implementation of these two processes.

4.4.1.1. Laws' texture energy

There are two major approaches to the analysis of textures in digital images; statistical and structural/syntactic methods (Maxwell et al., 2003).

Statistical methods analyse textures using spatial or frequency-based filters. Examples of spatial filters are Laws' texture energy filter and co-occurrence matrices. Frequency-based filters include Fourier, discrete cosine and wavelet transforms. Structural/syntactic methods break a texture into symbolic units and then represent the texture in terms of the relationships between elements. However, (Maxwell et al., 2003) notes that currently, statistical methods dominate the field of texture analysis.

As there is no clear consensus on the best statistical method (Randen et al., 1999), a simple approach for generating texture images is implemented in this thesis: Laws' texture energy filter (Laws, 1980a, Laws, 1979, Laws, 1980b). Constructing PBIs from PHIs is a comparatively simple texture edge detection problem; thus, implementing a simple approach like the Law's texture energy filter to perform this is sufficient.

Laws' texture energy filter (Laws, 1980a, Laws, 1979, Laws, 1980b) comprises a set of filters designed to identify specific primitive features in a local region within a digital image. The Laws' filter is focused on three vectors:

$$L_3 = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{4.3}$$

$$E_3 = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \tag{4.4}$$

$$S_3 = \begin{bmatrix} -1 & 2 & -1 \end{bmatrix} \tag{4.5}$$

L_3 is mnemonic for *Level* and its function is smoothing, E_3 is mnemonic for *Edge* and locates edges and S_3 is mnemonic for *Spot* and its purpose is to identify spots. Convolving these three vectors among themselves produces five additional vectors:

$$L_5 = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{4.6}$$

$$E_5 = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix} \tag{4.7}$$

$$S_5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix} \tag{4.8}$$

$$W_5 = \begin{bmatrix} -1 & 2 & 0 & -2 & 1 \end{bmatrix} \tag{4.9}$$

$$R_5 = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix} \tag{4.10}$$

The five vectors' names are mnemonics for *Level*, *Edge*, *Spot*, *Wave* and *Ripple*. Each represents the type of feature it is designed to detect. A set of 25 5×5 masks are obtained by multiplying the five vectors with themselves and one another. Twenty of these are transposes of another mask.

Applying the 25 masks to the PHI produces a set of 25 images where each image represents a texture plane. A windowing operation is performed on each of these 25 images, replacing each pixel with a Texture Energy Measure (TEM), ensuing in 25 Texture Energy Image (TEI) being produced. (Laws, 1980a, Laws, 1979, Laws, 1980b) showed that using a 15×15 moving window produced good results when computing the TEM, and that window size is adopted here. The TEM is computed using (4.11) (we follow the notation in (Kelly, 2010)):

$$TEI(x,y)=\sqrt{\sum_{i=-7}^{7}\sum_{j=-7}^{7}(x+i,y+j)^2} \tag{4.11}$$

The resulting TEI that are transposes of one another are then combined and are divided with $L_5\times L_5$, resulting in 14 texture planes being produced. The $L_5\times L_5$ image is used as the base image because it is the only image that does not produce a zero mean result. The 14 texture planes can be created by applying (4.12) which is norm used for contrast stretching in digital image(s) (we follow the notation in ((Fisher et al., 2003)), which is modified for this thesis):

$$T_{new}(x,y)=(T_{current}(x,y)-d)\left(\frac{a-b}{c-d}\right)+b \tag{4.12}$$

where $T_{new}(x,y)$ is the new texture image plane, $T_{current}(x,y)$ is the current texture image plane, *a* and *b* refers to the upper and lower limits, which are obtained from the base image, whereas, *c* and *d* refers to the highest pixel value and lowest pixel value, which are obtained from $T_{current}(x,y)$.

To ensure no filter dominates the edge magnitude computation, the 14 texture planes are normalised using (4.13)(we follow the notation in (Maxwell et al., 2003)):

$$\hat{T}_i(x,y)=\frac{T_i(x,y)-\hat{T}_i}{\sigma_i} \tag{4.13}$$

(Maxwell et al., 2003) notes that (4.13) gives each texture plane T_i equal significance by normalising by the mean $\hat{T}_i$ and standard deviation σ_i. The 14

Laws' texture feature images are obtained by restricting the range values to 4 standard deviations and scaled to a range $[0,255]$.

The Compass operator is then applied to these 14 Laws' texture features in order to obtain the PBI.

4.4.1.2. Compass operator

The Compass operator was originally intended (Ruzon et al., 1999) to perform colour edge detection by comparing two halves of an oriented circle at a variety of orientation for each pixel in an image. However, (Maxwell et al., 2003) argue that the Compass operator can also be used to perform texture edge detection.

The Compass operator is applied to each of the 14 Laws' texture feature images. A Compass operator is built on top of each pixel with a radius size of r. For every texture feature image, each pixel is evaluated at different orientations. Each of these different orientations splits the Compass operator into two halves which builds two histogram distributions. The two histogram distributions signify the (x, y)-pixel coordinate placement within one half of the two halves. (Maxwell et al., 2003) notes that each pixel within a texture image represents an area of the size of the underlying filter for a given plane, therefore, pixels that lie within the radius, r are ignored when computing the histogram distributions because these pixels represents a mixed or transitional distribution of texture feature rather than one of the distributions on either side of the edge. Consequently, the numbers of pixels that are ignored are ruled by the size of a mini radius, mr originated from r. Thus, all pixels lying within mr are not included in the histograms.

Once the histograms have been built, the magnitude of the differences between the two distributions is computed. (Maxwell et al., 2003) shows that this can be done using dynamic time warping (DTW) (Rabiner et al., 1993). Dynamic time warping is a dynamic programming approach used to find the optimal correspondence between two time varying sequences by measuring their similarities. (Maxwell et al., 2003) shows that DTW is equally applicable to one dimensional histograms.

After the Compass operator is applied to each texture feature image, the recorded maximal distance and orientation results for each plane are summed and averaged, generating a single average image. A maximum value is initially found by traversing through the average image. The edge image is then obtained via thresholding at a percentage of the maximum value. The PBI is generated after applying a thinning algorithm (Gonzalez et al., 2002) to the edge image.

4.4.2. Results

The texture edge detection algorithm was applied to the PHIs shown in Figure 4.1 and Figure 4.2.

Each Law's texture feature image was evaluated at four different orientations (0, 45, 90 and 135 degrees) by the Compass operator. A Compass operator with a radius, r size of 10 and mini radius, mr size of 5 is used for every PHI. These parameters were determined empirically. A bin size of 10 was selected when computing the histograms at each orientation, and thresholds of 70-75% of the maximum applied, following (Maxwell et al., 2003).

Figure 4.3 (a) and (b) demonstrates a view-based representation of PBIs generated for video(s) sequence in Chapter 3 (indicated by the figure number in brackets) where the target is being tracked while moving normally.

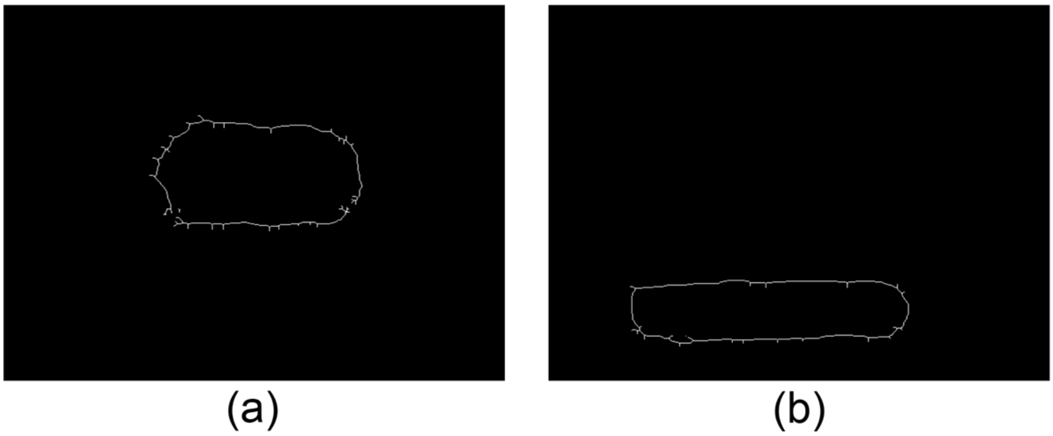


Figure 4.3 PBIs for a: (a) yellow circle moving normally (Figure 3.15); (b) tennis ball moving normally (Figure 3.19).

Figure 4.4 (a), (b), (c) and (d) show PBIs generated from video sequences in Chapter 3 (as indicated by the figure number in brackets) that exhibit a target experiencing occlusion. Figure 4.4 (e) shows the PBI generated when a target experienced self-occlusion as shown in Figure 3.29 in Chapter 3.

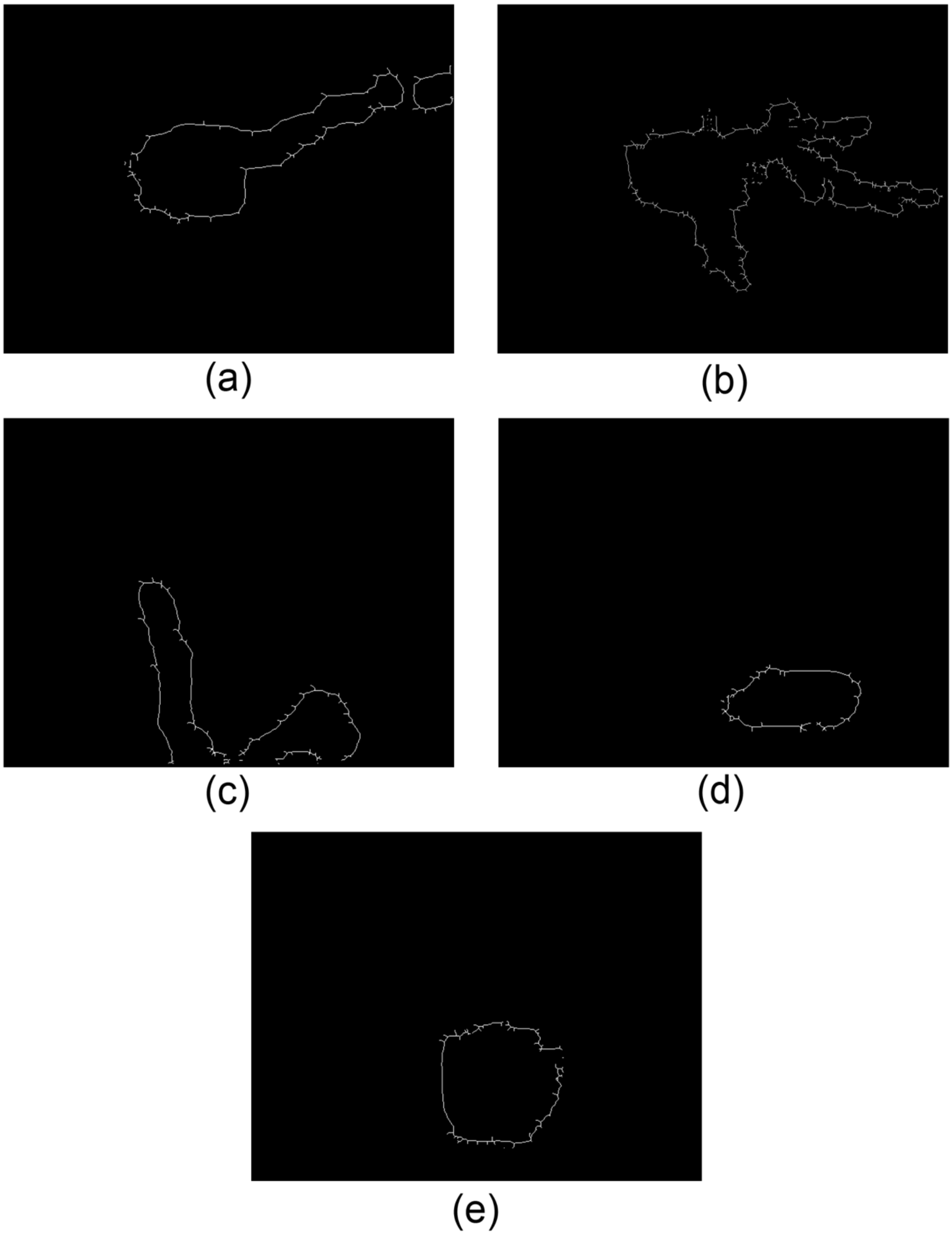


Figure 4.4 PHIs for a: (a) yellow circle experiencing occlusion (Figure 3.21); (b) yellow circle experiencing occlusion and in the presence of clutter (Figure 3.23); (c) tree occluding a woman (Figure 3.25); (d) tennis ball experiencing occlusion (Figure 3.27); (e) table tennis bat experiencing self-occlusion (Figure 3.29).

Figure 4.5 (a), (b), (c) and (d) illustrates PBIs produced for video sequences in Chapter 3 (as indicated by the figure number in brackets) that shows a target experiencing camouflage.

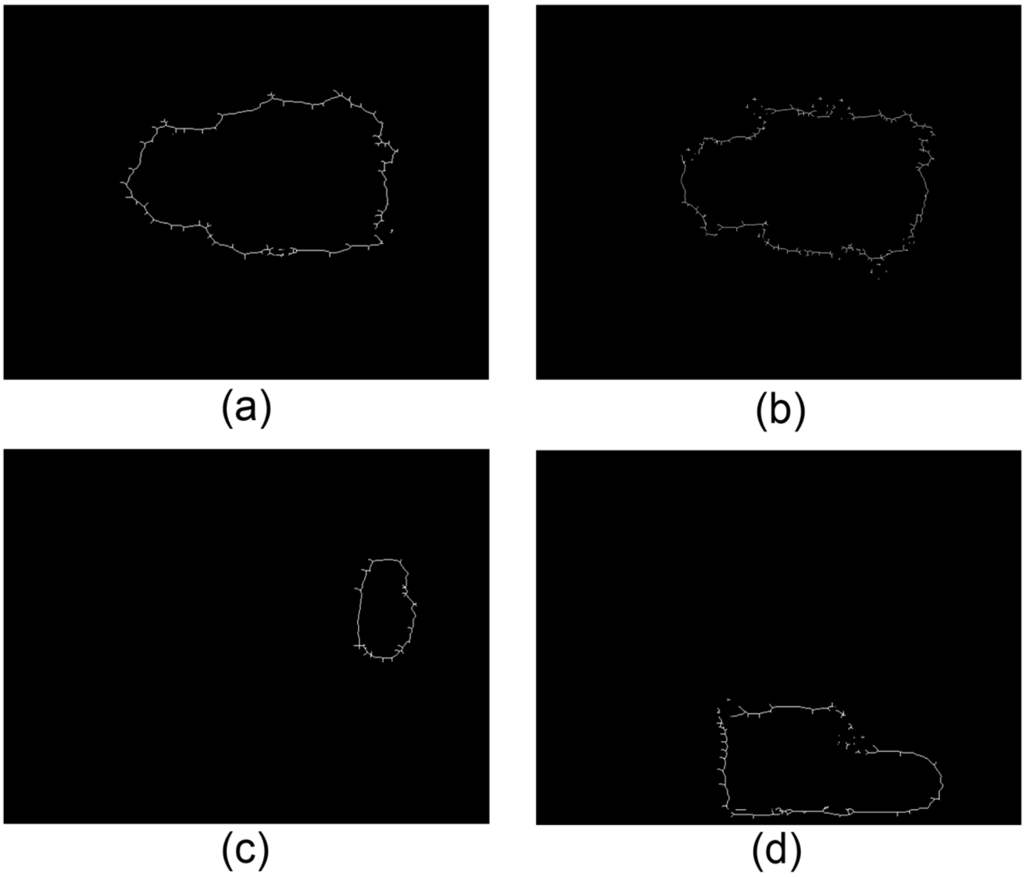


Figure 4.5 PBIs for a: (a) yellow circle experiencing camouflage (Figure 3.31); (b) yellow circle experiencing camouflage and in the presence of clutter (Figure 3.33); (c) player camouflaging a football (Figure 3.35); (d) tennis ball experiencing camouflage (Figure 3.37).

Applying a texture edge detection algorithm to the PHIs clearly captures potentially valuable information about the behaviour of particle sets during disruptive events. Section 4.5 presents two ways to extract this information, which is then exploited using a process-behaviour chart to detect occlusion and camouflage events.

4.5. Measuring particle spread

Results presented in Chapter 3 showed that the spread of the particle sets over a given temporal interval indicates whether occlusion or camouflage has taken place within that period. The PHI and PBI data structures provide a compact representation of the spatio-temporal behaviour of particle sets, and support a number of measures of particle distribution. Two such measures are investigated in what follows. One emphasises the temporal information available, the other the spatial.

4.5.1. Hypothesis

The method reported in Chapter 3 relies upon particle clusters formed, independently, at each time step. Though the approach was successful in detection occlusion and camouflage events, reliance on individual particle clusters raises some concerns. The spread of a given cluster is a complex function of the relationship between appearance model, target and background properties. Local variations in particle spread might be introduced by background clutter, illumination changes, etc., distorting the measurements upon which event detection is based. The PHI and PBI representations provide opportunities to measure particle spread at a given time point while taking into account particle behaviour over a longer time period. We hypothesise that this will provide a more robust measure of particle spread and so more reliable occlusion and camouflage detection.

4.5.2. Emphasising temporal behaviour

To exploit the temporal information in the PHI and PBI, we overlay the target path reported by the tracker.

The location of the particle set at a given time step is approximated by the mean particle location, $\overline{X}_t(\mu_x, \mu_y)$. In Chapter 3, this is provided by the EM algorithm. Here, however, it is computed using (4.14) and (4.15):

$$\mu_x = \frac{\sum_{i=1}^{N} x_i \Pi_i}{\sum_{i=1}^{N} \Pi_i} \tag{4.14}$$

$$\mu_y = \frac{\sum_{i=1}^{N} y_i \Pi_i}{\sum_{i=1}^{N} \Pi_i} \tag{4.15}$$

where x and y represent the particle's location, N is the sample population size and Π is the particle weight computed by the Condensation algorithm.

Mean particle locations over a specified temporal interval provide an indication of the tracker's estimate of target path (henceforth referred to as the tracker's path estimate). Particle spread is computed by overlaying the tracker's path estimate on the PBI. However, mean particle locations can be noisy, as a result of fluctuations in particle behaviour during tracking. Thus, the tracker's path estimates are improved by applying Gaussian smoothing prior to computing the particle spread. Gaussian smoothing is implemented as detailed in (Trucco et al., 1998).

A line normal to the tracker's path estimate is computed for each mean particle location and used to estimate the particle spread. A vector $(\hat{x}, \hat{y})$ between two mean particle(s) on the tracker's estimate path is computed using (4.16) and (4.17):

$$\hat{x} = \mu_{x_t} - \mu_{x_{t-1}} \tag{4.16}$$

$$\hat{y} = \mu_{y_t} - \mu_{y_{t-1}} \tag{4.17}$$

where μ_x and μ_y represent the mean particle locations and t is the time step. A normal line in the directions of $(-\hat{y}, \hat{x})$ and $(\hat{y}, -\hat{x})$ is then computed from $(\hat{x}, \hat{y})$ and plotted using the Bresenham line algorithm (Bresenham, 1965). Bresenham's algorithm allows us to navigate along the normal line, in both directions, from $(\mu_{x_{t-1}}, \mu_{y_{t-1}})$ until a boundary point (x_b, y_b) in the PBI is reached. The distance between $(\mu_{x_{t-1}}, \mu_{y_{t-1}})$ and (x_b, y_b) is then computed using (4.18):

$$d(x,y) = \sqrt{\left(\mu_{x_{t-1}} - x_b\right)^2 + \left(\mu_{y_{t-1}} - y_b\right)^2} \tag{4.18}$$

The distance measured in both directions is summed to estimate particle spread.

The process is repeated for every mean particle location on the tracker's path estimate. Graphical representations of the results of computing the particle spread in this way are shown as green lines and blue dots overlaid on the PBI. The green lines represent the normal lines, while the blue dot(s) represent mean particle locations.

Once particle spread is computed for each point on the tracker's path estimate, the resulting data is input to a process-behaviour chart to detect the occlusion and camouflage events. The need to construct PHI and PBIs before applying the process-behaviour chart introduces a delay in the detection of disruptive events – detection can only occur at the end of the time interval used to build the PHI. It is hypothesised, however, that the increased robustness of the method will compensate for this short delay.

Gaussian smoothing with a standard deviation, σ, of 2 is used to reduce noise in the tracker's path estimate. A standard deviation of 2 was selected after trial and error using standard deviations ranging from 1 to 5 showed that using a standard deviation value greater than 2 causes the mean particle location on the tracker's estimate path to experience high fluctuations if the path is too short compared to the size of the Gaussian mask.

The process-behaviour chart analysis in Chapter 3 shows that a moving window size of 7 produces good results. As a result, and to allow comparison of methods, the same moving window size is used throughout this analysis.

4.5.2.1. Normal tracking

Figure 4.6 and Figure 4.8 demonstrate particle spread measurement based on temporal behaviour for video sequences in which the target is moving normally. The process-behaviour chart graphs are shown in Figure 4.7 and Figure 4.9 respectively.

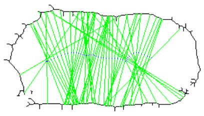


Figure 4.6 Particle spread, measured relative to the tracker path, for a yellow circle moving normally.

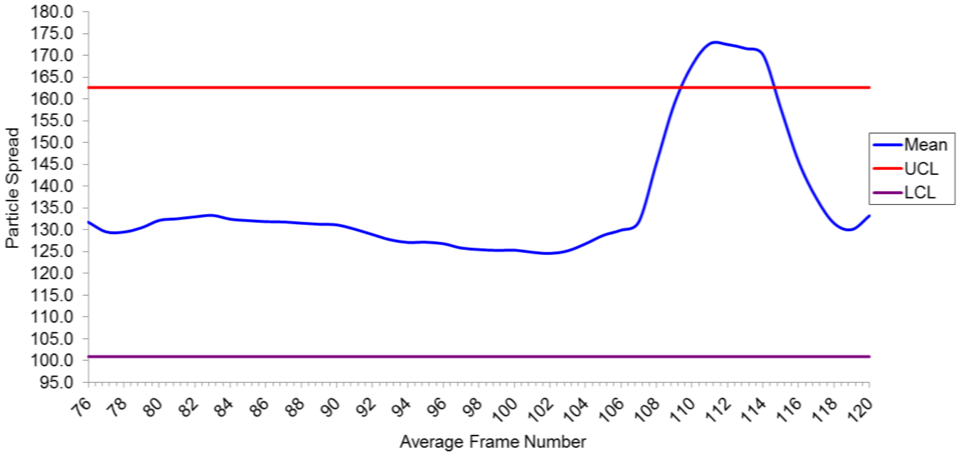


Figure 4.7 Shewhart control chart analysing particle spread for a yellow circle moving normally.

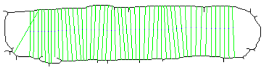


Figure 4.8 Particle spread, measured relative to the tracker path, for a tennis ball moving normally.

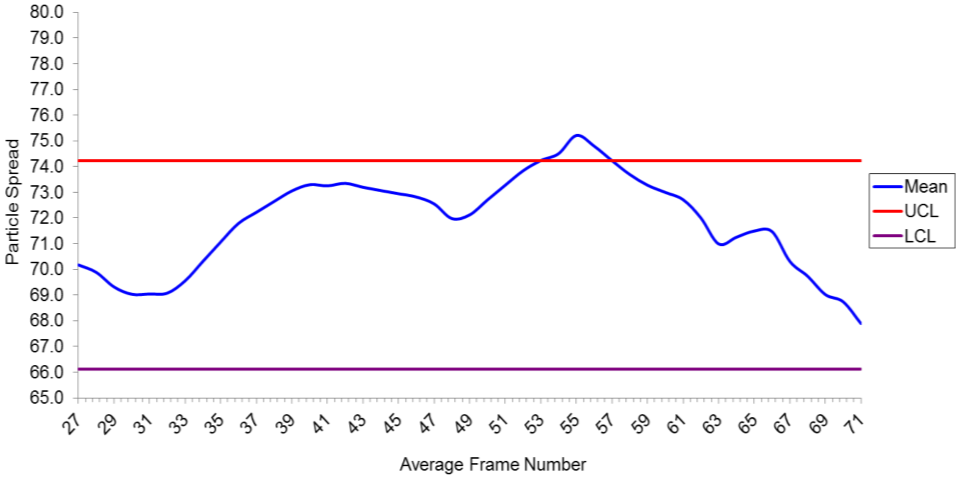


Figure 4.9 Shewhart control chart analysing particle spread for a tennis ball moving normally.

During normal tracking:

- The tracker's path estimate for a tennis ball moving normally (Figure 4.8) is smoother than the path estimates for a yellow circle moving normally (Figure 4.6).
- The Shewhart control charts in Figure 4.7 and Figure 4.9 both show the control chart mean exceeding the upper control limit momentarily. While the exceeding period is too short to cause the tracker to fire in the event monitored in Figure 4.9, the event monitored in Figure 4.7 does result in the tracker firing incorrectly. The false positive results in both figures occur due to noise in the tracker's estimate of target path.

4.5.2.2. Occlusion

Measurement of particle spread relative to the trackers path estimate from video sequences showing occlusion is demonstrated in Figure 4.10, Figure 4.12, Figure 4.14 and Figure 4.16. Figure 4.11, Figure 4.13, Figure 4.15 and Figure 4.17 give corresponding process-behaviour chart graphs. Figure 4.18 and Figure 4.19 show results obtained from a self-occluding target.

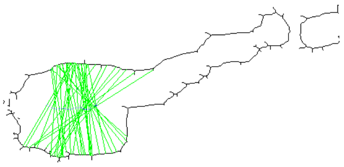


Figure 4.10 Particle spread for a yellow circle experiencing occlusion.

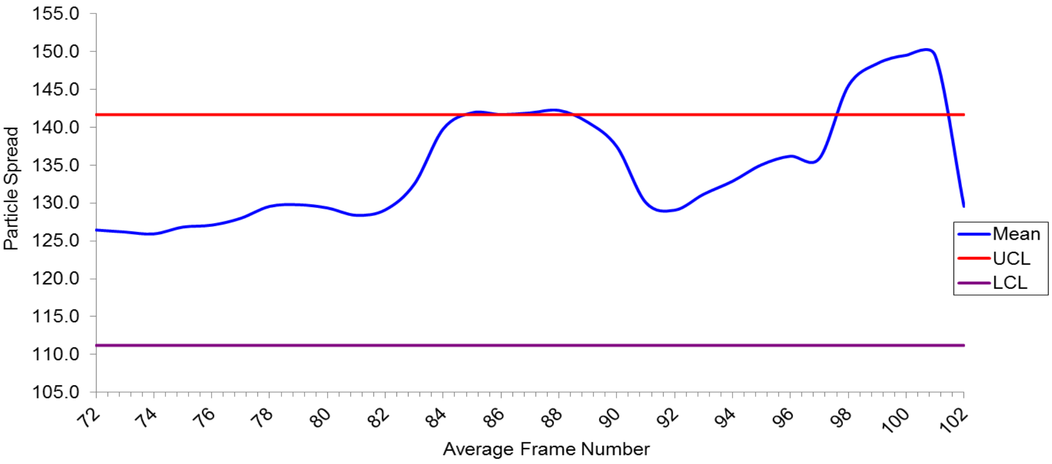


Figure 4.11 Shewhart control chart analysing particle spread for a yellow circle experiencing occlusion.

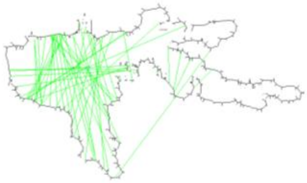


Figure 4.12 Particle for a yellow circle experiencing occlusion and with clutter present.

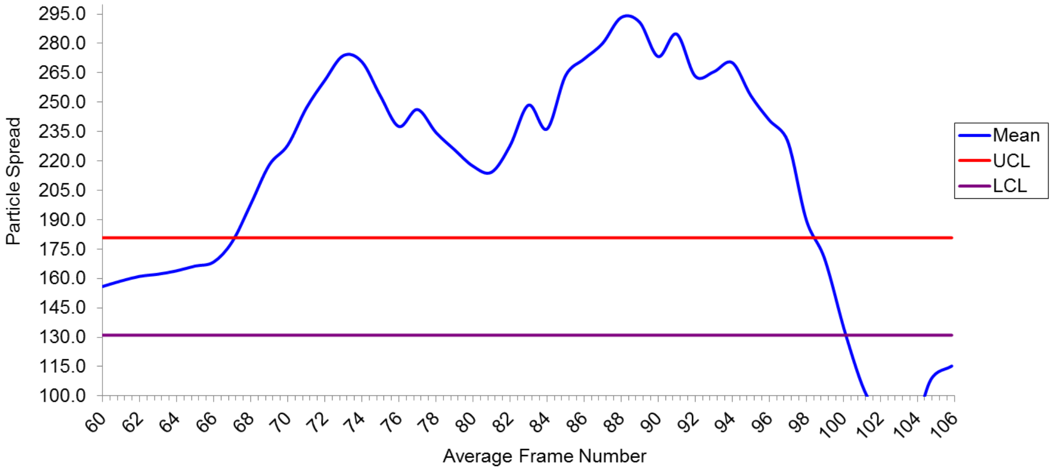


Figure 4.13 Shewhart control chart analysing particle spread for a yellow circle experiencing occlusion with clutter present.

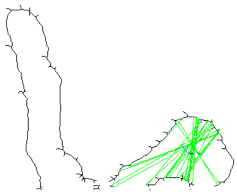


Figure 4.14 Particle spread for a tree occluding a woman.

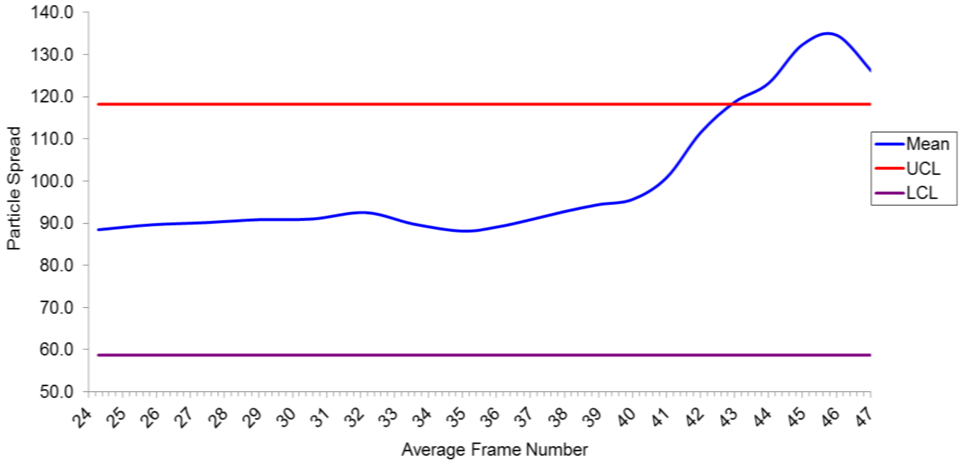


Figure 4.15 Shewhart control chart analysing particle spread for a tree occluding a woman.

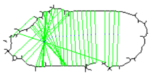


Figure 4.16 Particle spread for a tennis ball experiencing occlusion.

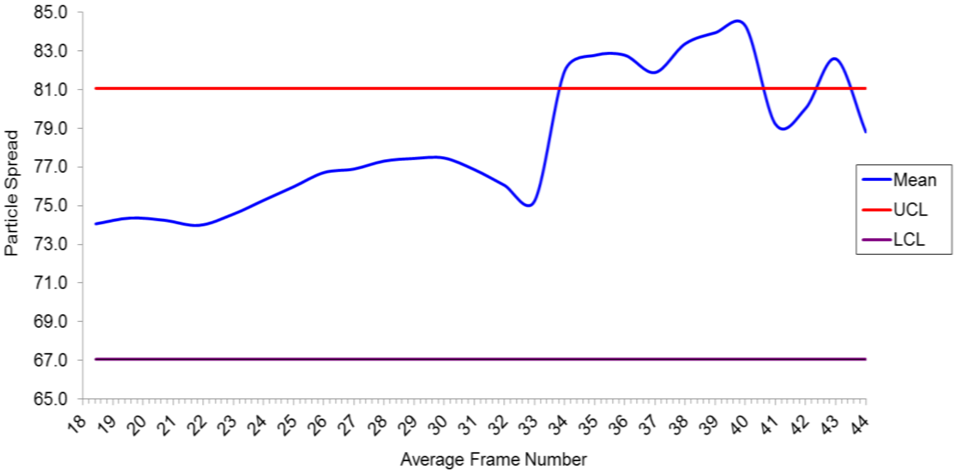


Figure 4.17 Shewhart control chart analysing particle spread for a tennis ball experiencing occlusion.

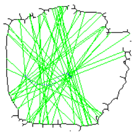


Figure 4.18 Particle spread for a table tennis bat experiencing self-occlusion.

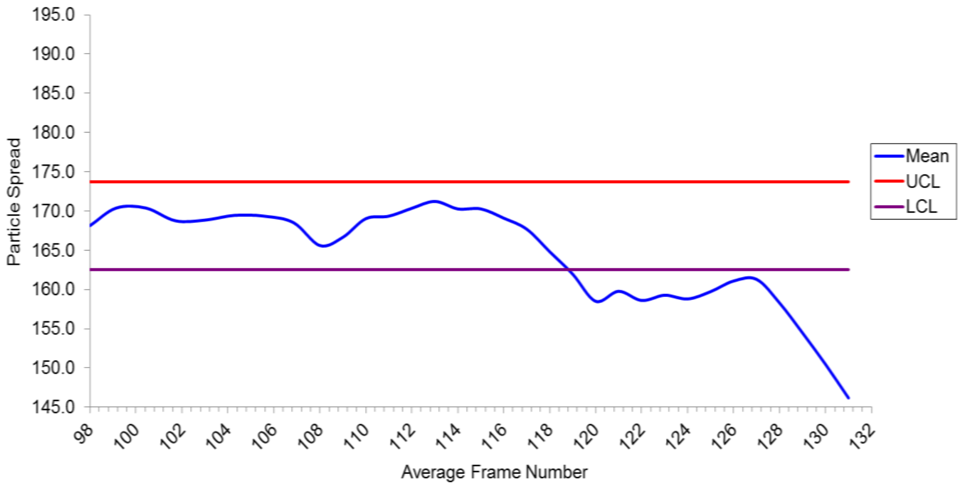


Figure 4.19 Shewhart control chart analysing particle spread for a table tennis bat experiencing self-occlusion.

In the event of occlusion:

- A relatively steady and smooth tracker's path estimate is recorded prior to the occurrence of occlusion (as shown in Figure 4.16). However, as occlusion occurs, the path starts to fluctuate and begins deviating away from the earlier steady path (Figure 4.10 and Figure 4.16 shows blue dots deviating away from the path as occlusion happens). As the target is lost, the path estimates becomes too noisy as at this stage the particles may be drifting within the image plane trying to recapture the target once it reappears or tracking clutter (Figure 4.12) which has similar characteristics to the target. As a result, considerable noise is introduced into the orientations of the measurements made, and the resulting data does not capture the reduction in particle spread that is clearly seen in Figure 4.10 and Figure 4.14.

- The Shewhart control charts in Figure 4.11, Figure 4.13, Figure 4.15 and Figure 4.17 shows the control mean to be within acceptable limits during normal tracking, when the tracker's path estimate is steady. However, in the event of occlusion, all four charts show the control mean exceeding the upper control limit. This occurs because the path estimate recorded after occlusion is too noisy and the shrunk regions in the PBI as a result of occlusion are not measured.

- In the event of self-occlusion (Figure 4.18), the tracker's path estimate is seen steady (analysing the blue dots from left to right). Though, the blue dots starts fluctuating and overlapping over the earlier steady path (on the right side of the path). This may imply that self-occlusion is starting to occur. Shewhart control chart in Figure 4.19 shows the control mean falling below the lower control limit as a result of self-occlusion.

4.5.2.3. Camouflage

Figure 4.20, Figure 4.22, Figure 4.24 and Figure 4.26 show the computation of particle spread based on relative to the tracker’s path estimate for video sequences in which targets experience camouflage during tracking. Figure 4.21, Figure 4.23, Figure 4.25 and Figure 4.27 shows corresponding process-behaviour chart graphs.

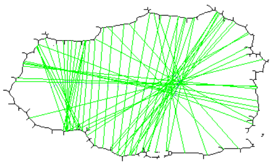


Figure 4.20 Particle spread for a yellow circle experiencing camouflage.

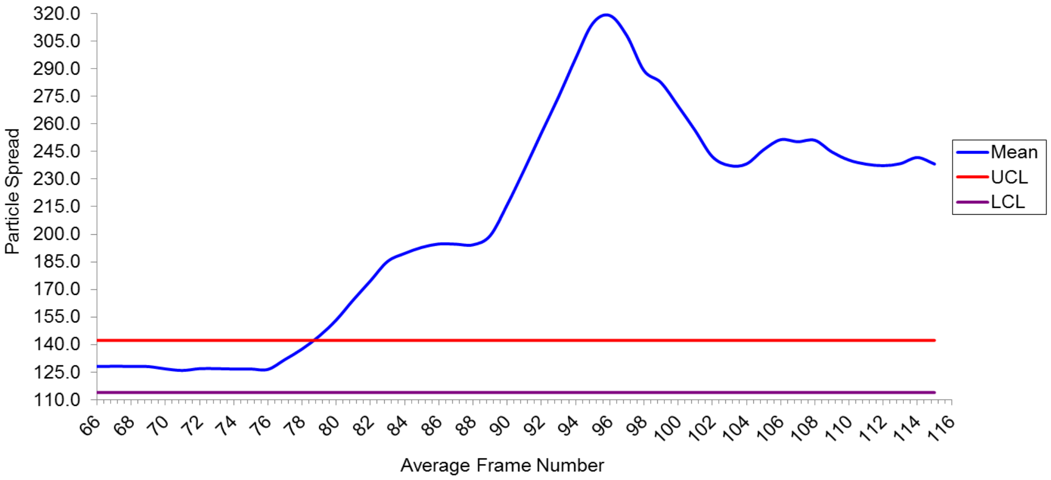


Figure 4.21 Shewhart control chart analysing particle spread for a yellow circle experiencing camouflage.

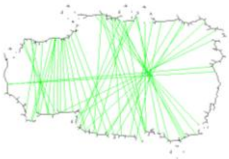


Figure 4.22 Particle spread for a yellow circle experiencing camouflage and in the presence of clutter.

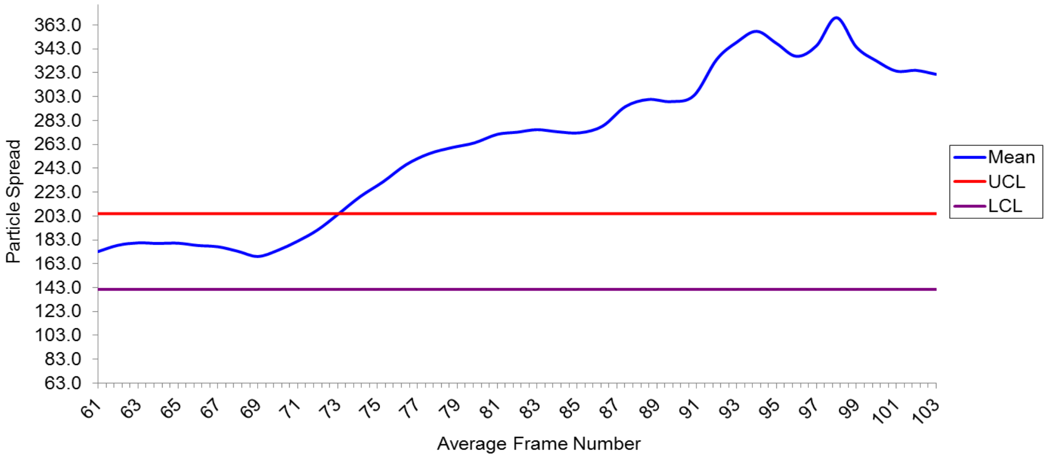


Figure 4.23 Shewhart control chart analysing particle spread for a yellow circle experiencing camouflage and with clutter present.

Figure 4.24 Particle spread for a player camouflaging a football.

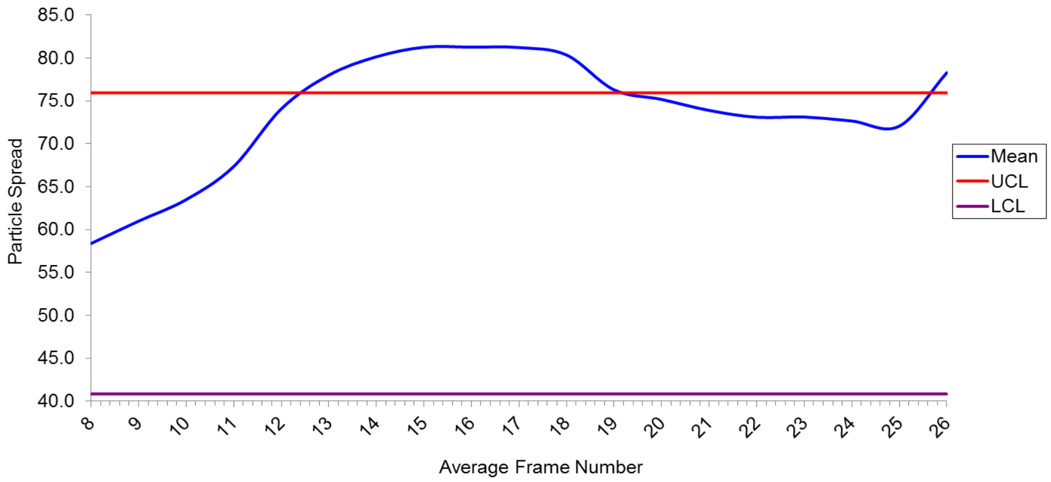


Figure 4.25 Shewhart control chart analysing particle spread for a player camouflaging a football.

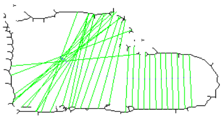


Figure 4.26 Particle spread for a tennis ball experiencing camouflage.

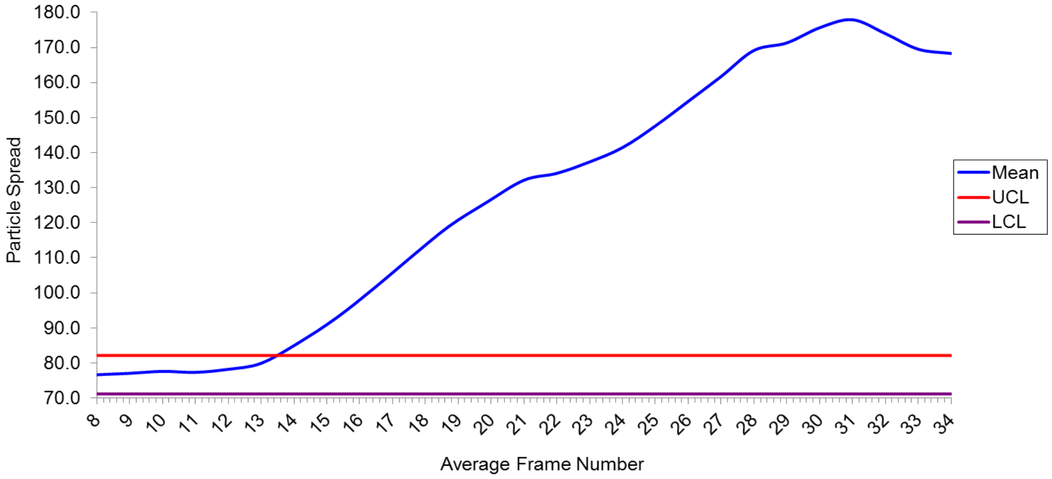


Figure 4.27 Shewhart control chart analysing particle spread for a tennis ball experiencing camouflage.

In the event of camouflage:

- As the target is being tracked moving normally (prior to the occurrence of camouflage), a relatively steady and smooth tracker's path estimate is recorded. However, as camouflage occurs, the path seems to experience some form of fluctuation (Figure 4.20). Additionally, some of the path estimates start to overlap the preceding path estimates (Figure 4.22, Figure 4.24 and Figure 4.26). This occurs from the particles moving aggressively within the camouflaging object, causing a noisy path estimate to be recorded. As a result, the direction in which the particle spread is measured varies. This can be clearly seen in Figure 4.24 where two particle spread is measured horizontally as opposed to vertically.

- Shewhart control charts in Figure 4.21, Figure 4.23, Figure 4.25 and Figure 4.27 all show the control mean exceeding the upper control limit in the event of camouflage.

4.5.2.4. Summary

Analysis showed that six false positives (refer to Figure 4.7, Figure 4.9, Figure 4.11, Figure 4.13, Figure 4.15 and Figure 4.17) were recorded for the Shewhart control chart when analysing normal tracking and occlusion events. These false positives occurred due to the less smooth and noisy path recorded. Analysis also showed the control chart recorded five true positives when analysing self-occlusion and all camouflaging events. It should be noted here that the gaps of the PBI played no role in contributing to these results.

Though the tracker’s path estimate successfully formed the basis of instantaneous particle spread measurements in Chapter 3, its use to measure the width of the occupied region of a PHI does not appear to be viable. Therefore, in section 4.5.3, an alternative approach is examined.

4.5.3. Emphasising spatial behaviour

This section presents an alternative approach to the computation of particle spread from a PBI. The emphasis here is on spatial information, specifically the medial axis of the occupied region of the PHI. The PBIs are pre-processed using image processing methods to estimate the medial axis. The medial axis is then used to estimate particle spread as before.

Three test videos are selected at random to illustrate the method. The selected examples are: (1) a yellow circle moving normally but in the presence of clutter; (2) a yellow circle experiencing occlusion and (3) a tennis ball experiencing camouflage.

4.5.3.1. Pre-processing

Prior to applying image processing methods to the PBIs, the non-thinned images generated by the Compass operator are manually edited to close any gaps in the PBI's boundary. This could be attempted automatically, but consideration of this is beyond the scope of this thesis. Figure 4.28(a), Figure 4.29(a) and Figure 4.30(a) show the non-thinned PBIs before manual editing, while Figure 4.28(b), Figure 4.29(b) and Figure 4.30(b) show the same structures post-editing.

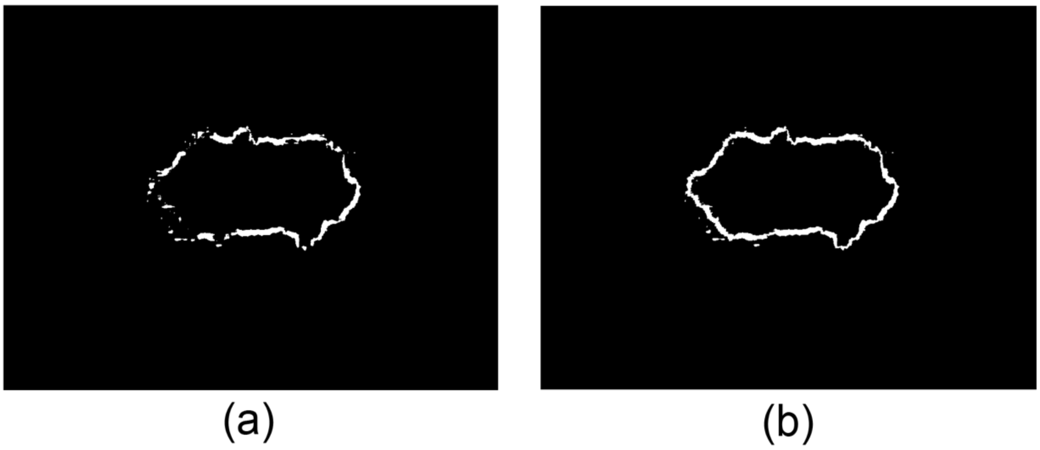


Figure 4.28 Non-thinned PBIs for a yellow circle moving normally but in the presence of clutter: (a) before manual edit; (b) after manual edit.

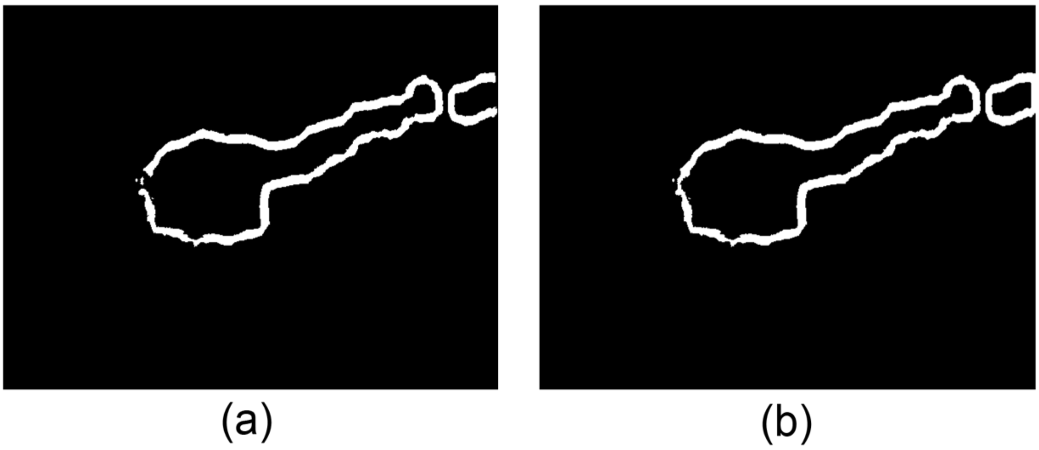


Figure 4.29 Non-thinned PBIs for a yellow circle experiencing occlusion: (a) before manual edit; (b) after manual edit.

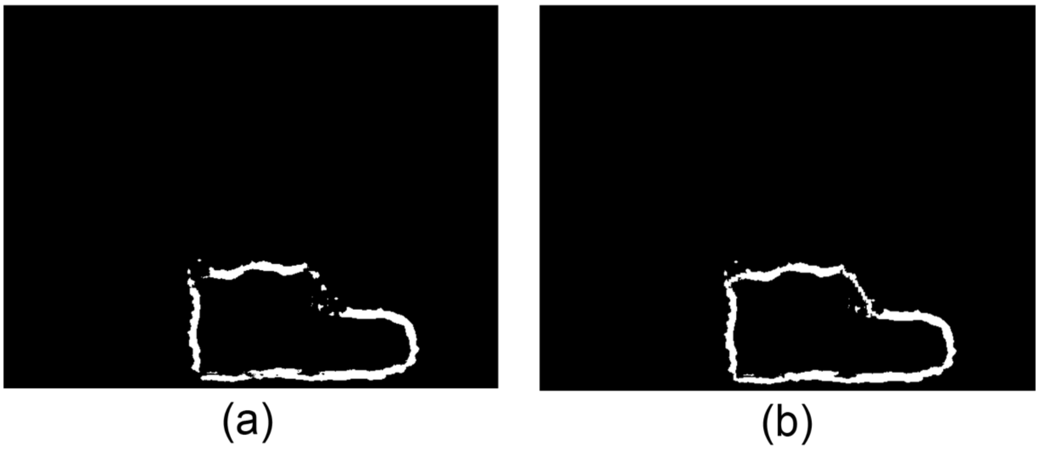


Figure 4.30 Non-thinned PBIs for a tennis ball experiencing camouflage: (a) before manual edit; (b) after manual edit.

The improved PBIs are then thinned to produce single pixel wide boundaries using the thinning algorithm implemented in the Compass operator. The results are illustrated in Figure 4.31 (a), (b) and (c).

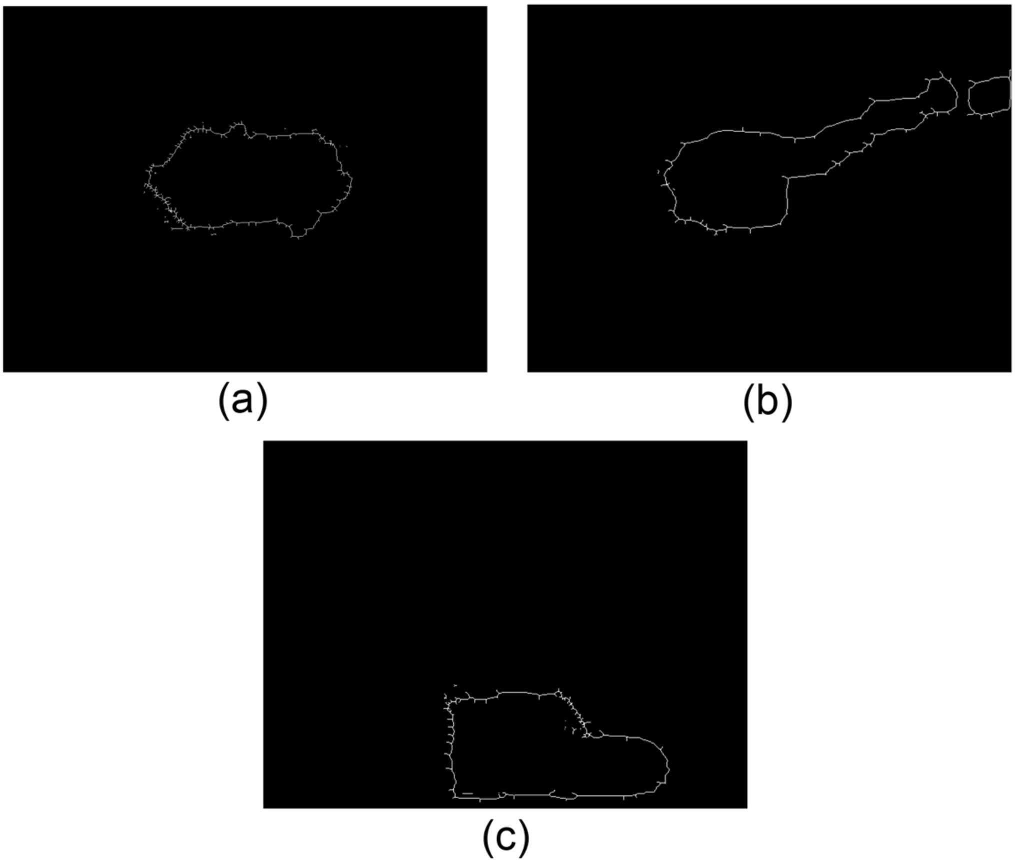


Figure 4.31 Thinned PBIs for: (a) a yellow circle moving normally but in the presence of clutter; (b) a yellow circle experiencing occlusion; (c) a tennis ball experiencing camouflage.

A flood fill algorithm (Foley et al., 1995, Hearn et al., 2004) is then applied to the thinned PBIs to fill up the inner region of the PBIs. The results are shown in Figure 4.32 (a), (b) and (c).

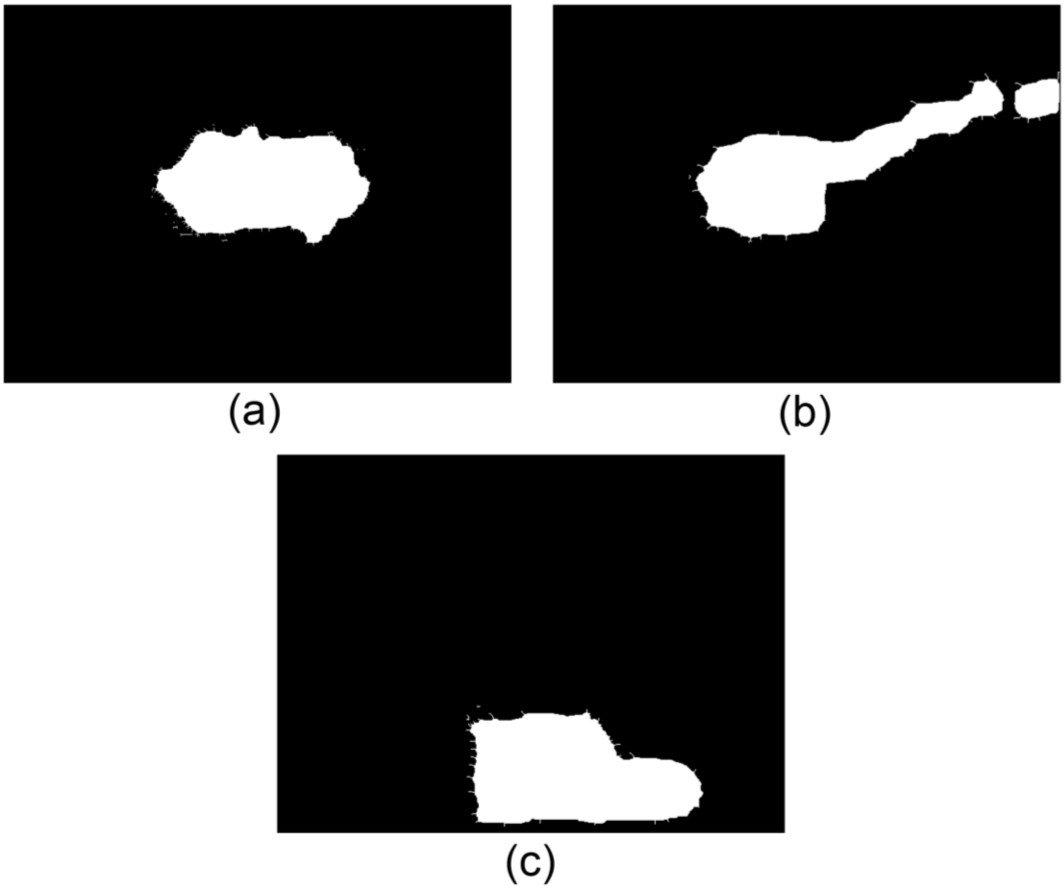


Figure 4.32 Flood-filled PBIs for: (a) a yellow circle moving normally but in the presence of clutter; (b) a yellow circle experiencing occlusion; (c) a tennis ball experiencing camouflage.

A distance transform (Borgefors, 1986) is applied to the flood filled PBIs to extract the medial axis. Distance transforms provide the shortest distance from each pixel to the closest boundary via the use of a distance map. A Chamfer distance transform approach is applied here, whereby a distance map is built by traversing the image in two passes. A filter mask is overlaid on each pixel during each pass. Trial and error using different filter mask sizes (e.g. 3×3 and5×5) showed that a filter mask size of 3×3 produces good results. Therefore, a filter mask size of 3×3 is used throughout this analysis. The distance transform results are shown in Figure 4.33 (a), (b) and (c).

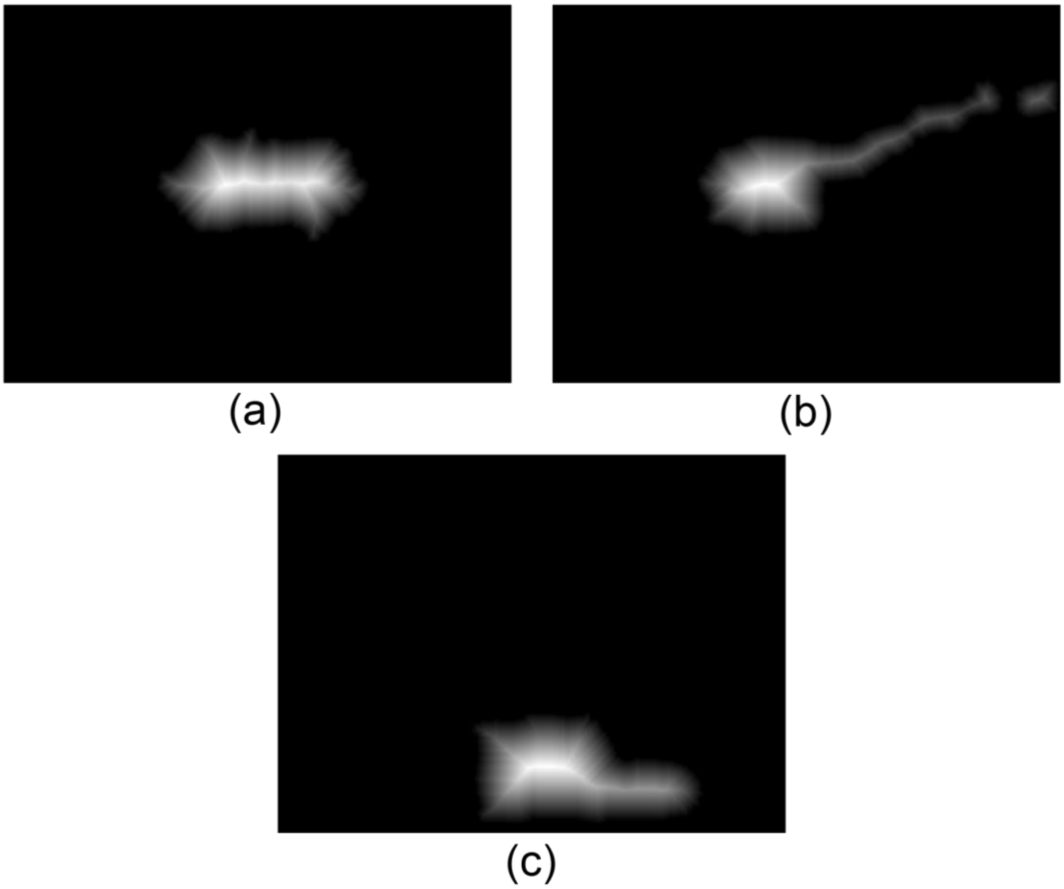


Figure 4.33 Distance transform results for: (a) a yellow circle moving normally but in the presence of clutter; (b) a yellow circle experiencing occlusion; (c) a tennis ball experiencing camouflage.

The medial axis is obtained by marking pixel coordinates whose distance value is the maximum within their 8-neighborhood. Medial axis results are shown in Figure 4.34 (a), (b) and (c).

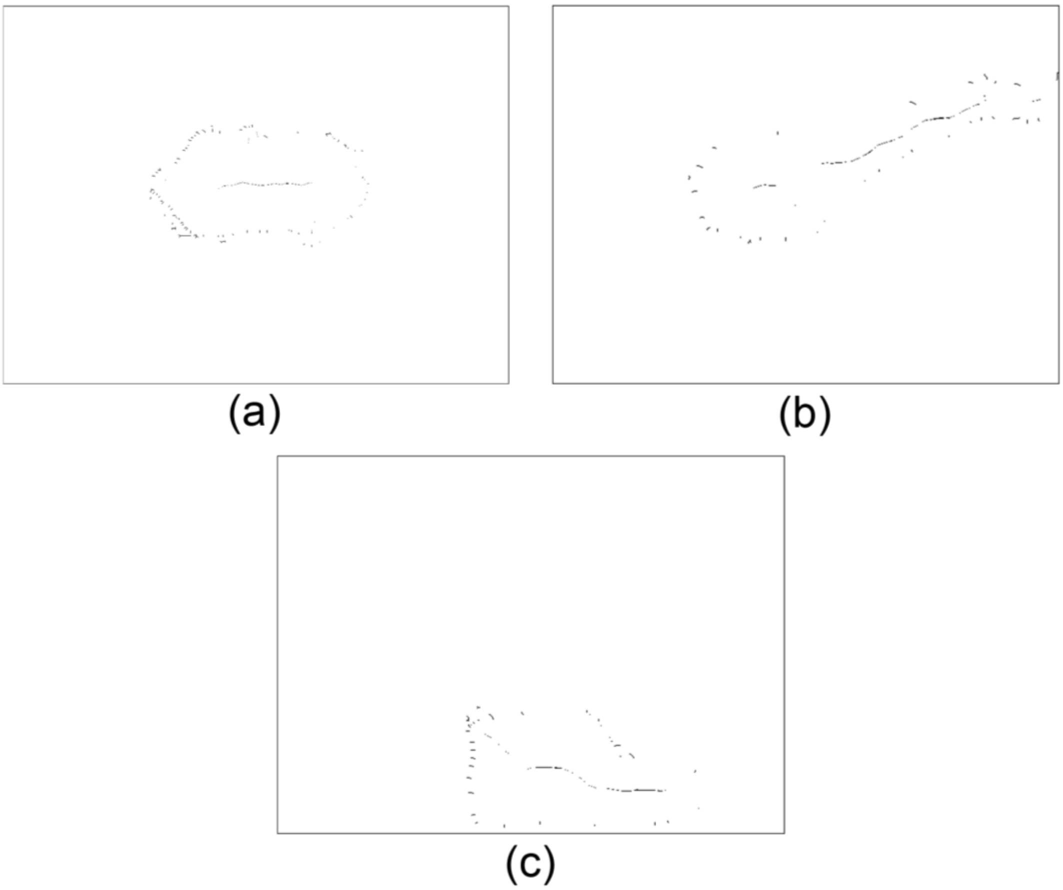


Figure 4.34 Medial axis results for: (a) a yellow circle moving normally but in the presence of clutter; (b) a yellow circle experiencing occlusion; (c) a tennis ball experiencing camouflage.

Examination of Figure 4.34 shows that in addition to the true medial axis, some additional features are recorded around the boundary of the occupied region. These are filtered out by inverting each medial axis result, and superimposing each one over their respective non-thinned PBIs. All "medial axis" points that fall upon boundary pixel coordinates in the non-thinned PBIs are removed. Figure 4.35 (a), (b) and (c) shows the final medial axes, after being inverted back to their original appearance. To provide a clearer view of these medial axes, Figure 4.35 (a), (b) and (c) shows an enlarged view of the medial axes from its original size.

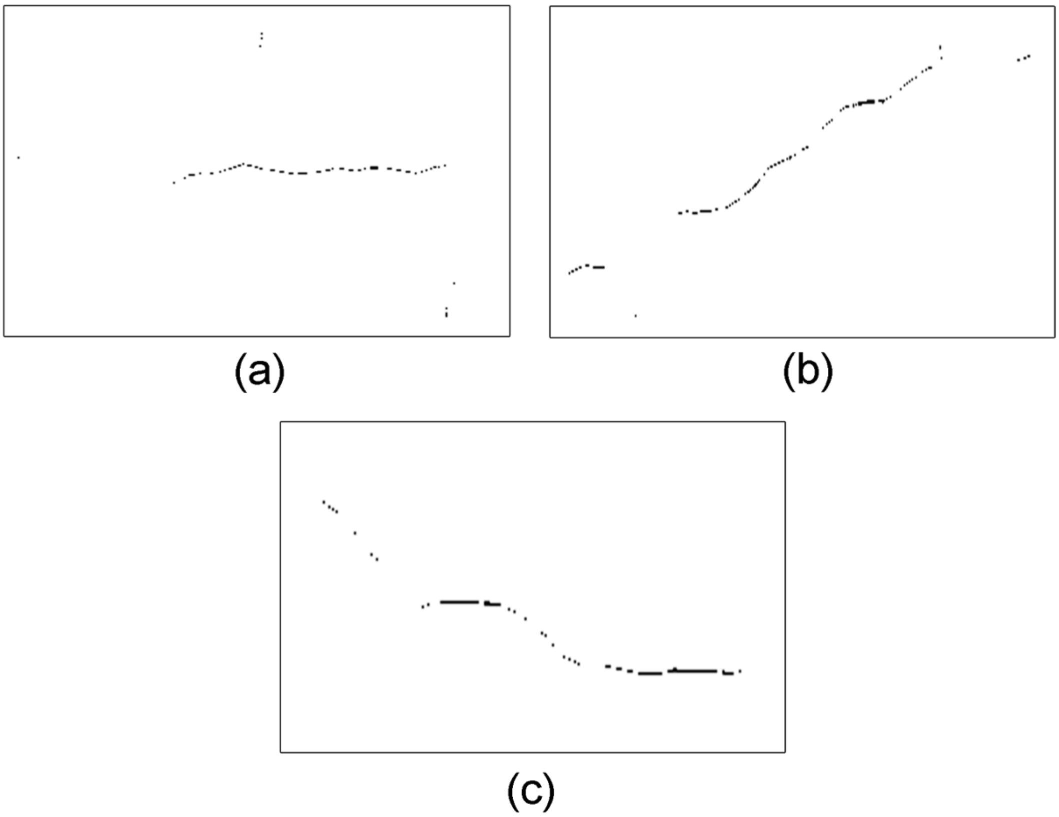


Figure 4.35 Filtered medial axis results for a: (a) yellow coloured circle moving normally but with the presence of clutter; (b) yellow coloured circle experiencing occlusion; (c) tennis ball experiencing camouflage.

The Bresenham line algorithm is then applied to each filtered medial axis result to produce a solid medial axis path by filling gaps between the medial axis points. Prior to this, any outlying points that are clearly not on the true medial axis are removed via manual edit. Although this could be attempted automatically, this is beyond the scope of this thesis. Applying the Bresenham line algorithm causes some parts of the medial axis path to become more than one pixel thick. This occurs because the optimal medial path also consists of points which are neighbours to a point located along the path. A thinning algorithm is therefore applied to produce a final medial axis path which is one pixel thick. The final medial axes are shown in Figure 4.36 (a), (b) and (c).

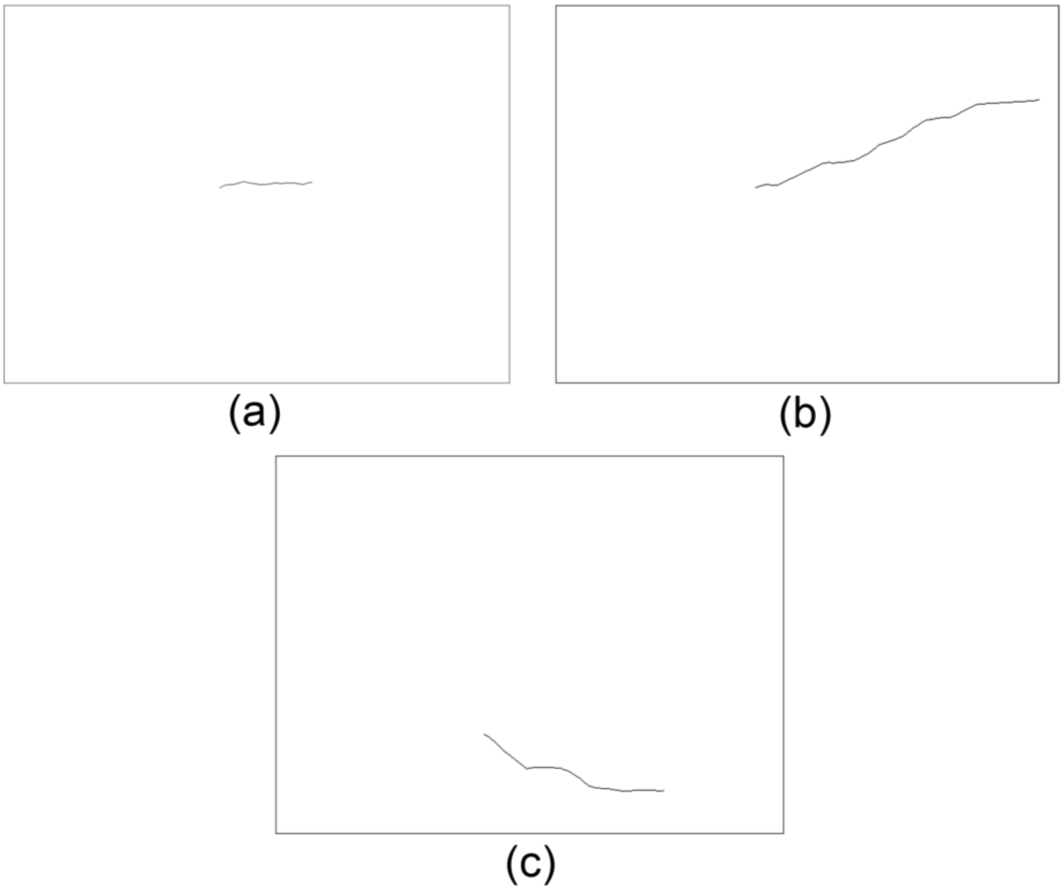


Figure 4.36 Final medial axes for: (a) a yellow circle moving normally but in the presence of clutter; (b) a yellow circle experiencing occlusion; (c) a tennis ball experiencing camouflage.

4.5.3.2. Implementation

During the computation of the particle spread as described in section 4.5.2, the direction in which the tracker's path is traversed is known. When replacing the tracker's path estimate with the potentially more stable medial axis, some method is needed of determining in which order measurements tied to medial axis points should be input to the process-behaviour chart. To allow the medial axis to be traversed in the correct direction, the distance is measured from the first and last points on the medial axis to the first point on the tracker's path estimate. The medial axis endpoint the shortest distance from the tracker starting point is taken as the start point for subsequent analysis.

A line normal to the medial axis is constructed for each point on the medial axis and used to estimate local particle spread as in section 4.5.2. Equations (4.16), (4.17) and (4.18) are, however, replaced with (4.19), (4.20) and (4.21):

$$\hat{x} = x_p - x_{p-1} \tag{4.19}$$

$$\hat{y} = y_p - y_{p-1} \tag{4.20}$$

$$d(x,y)=\sqrt{\left(x_{p-1}-x_b\right)^2+\left(y_{p-1}-y_b\right)^2} \qquad (4.21)$$

where x and y refer to the medial axis point coordinate and p is the medial axis point number on the path.

As before, in the figures below, green lines mark the normals along which spread is measured. Blue lines represent the medial axis. The particle spread results are used by the process-behaviour chart to detect occlusion and camouflage events. A moving window size of 7 is used throughout this analysis, again as before, and to allow comparison of results.

4.5.3.3. Normal tracking

Results of computing particle spread from the PBI medial axis for video sequences showing a target being tracked moving normally are shown in Figure 4.37 and Figure 4.39. Figure 4.38 and Figure 4.40 respectively show corresponding process-behaviour chart graphs.

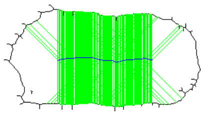


Figure 4.37 Particle spread computed from the PBI medial axis for a yellow circle moving normally.

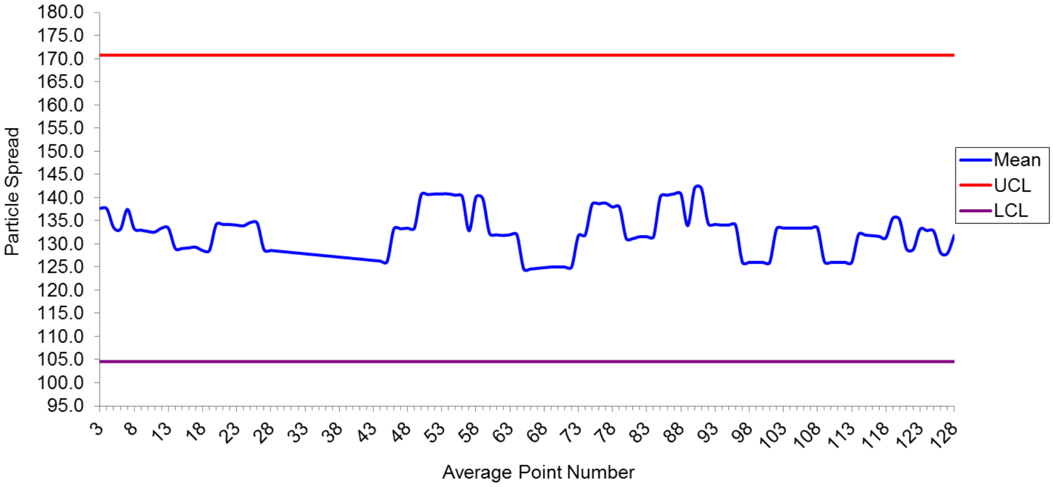


Figure 4.38 Shewhart control chart analysing particle spread computed from the PBI medial axis for a yellow circle moving normally.

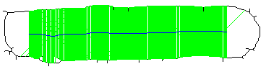


Figure 4.39 Particle spread computed from the PBI medial axis for a tennis ball moving normally.

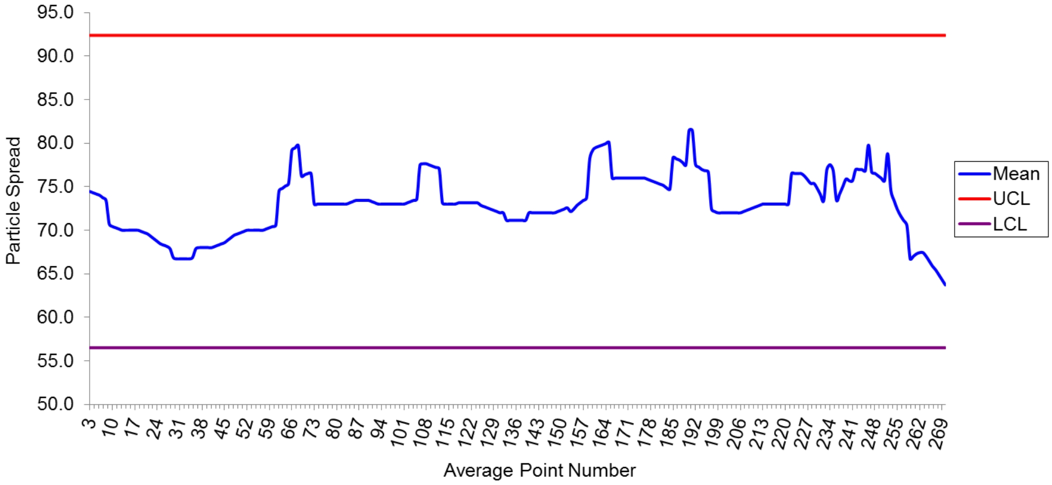


Figure 4.40 Shewhart control chart analysing particle spread computed from the PBI medial axis for tennis ball moving normally.

When normal tracking, examination of the particle spread data and process-behaviour charts show that:

- The PBI medial axis is smoother than the tracker's path estimate. This contributes to a better measurement of the width of the particle spread, as shown in Figure 4.37 and Figure 4.39.
- Shewhart control charts in Figure 4.38 and Figure 4.40 both record true positives.

4.5.3.4. Occlusion

Figure 4.41, Figure 4.43, Figure 4.45 and Figure 4.47 show estimates of particle spread computed from the PBI medial axis for video sequences showing a target experiencing occlusion. Process-behaviour chart graphs generated from this data are shown in Figure 4.42, Figure 4.44, Figure 4.46 and Figure 4.48. Figure 4.49 shows particle width data obtained from a target undergoing self-occlusion.

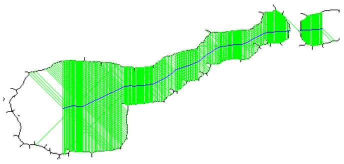


Figure 4.41 Particle spread computed from the PBI medial axis for a yellow circle experiencing occlusion.

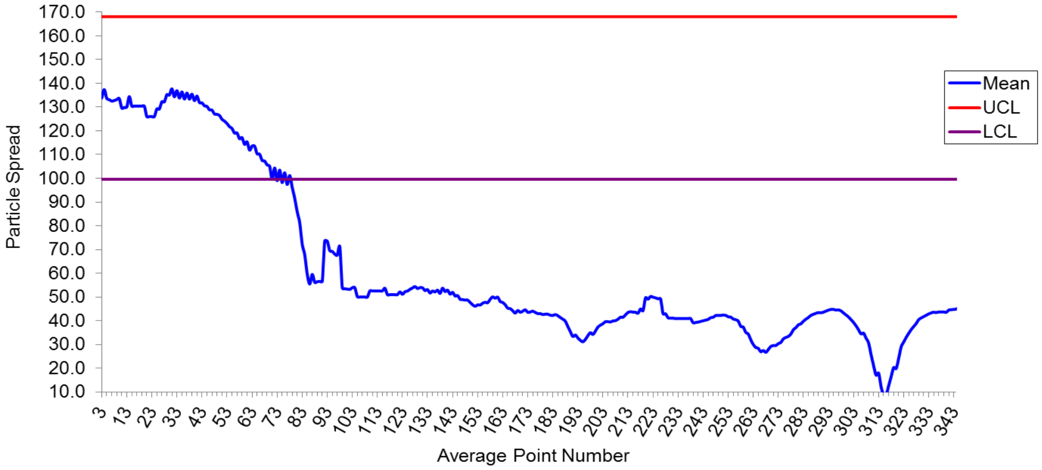


Figure 4.42 Shewhart control chart analysing particle spread computed from the PBI medial axis for a yellow circle experiencing occlusion.

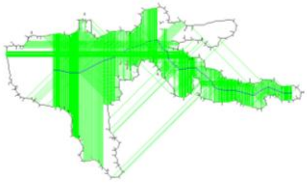


Figure 4.43 Particle spread computed from the PBI medial axis for a yellow circle experiencing occlusion and in the presence of clutter.

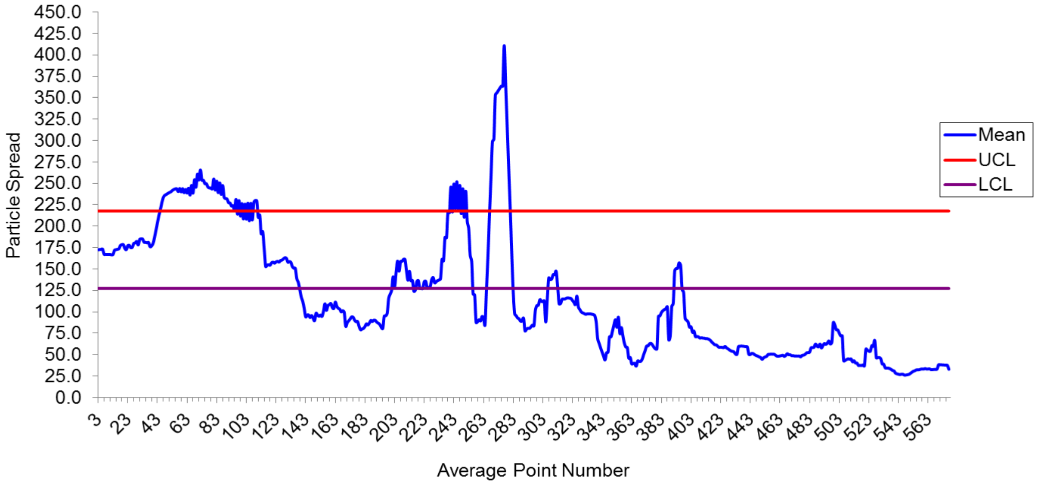


Figure 4.44 Shewhart control chart analysing particle spread computed from the PBI medial axis for a yellow circle experiencing occlusion and with clutter present.

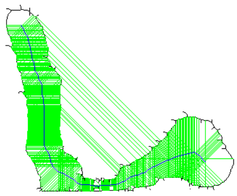


Figure 4.45 Particle spread computed from the PBI medial axis when a tree occludes a woman.

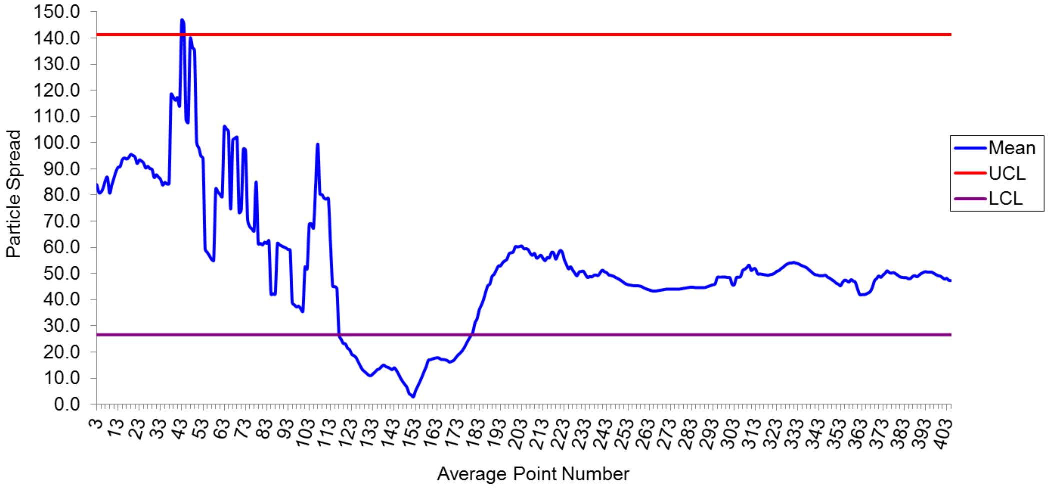


Figure 4.46 Shewhart control chart analysing particle spread computed from the PBI medial axis for a tree occluding a woman.

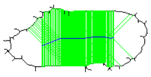


Figure 4.47 Particle spread computed from the PBI medial axis for a tennis ball experiencing occlusion.

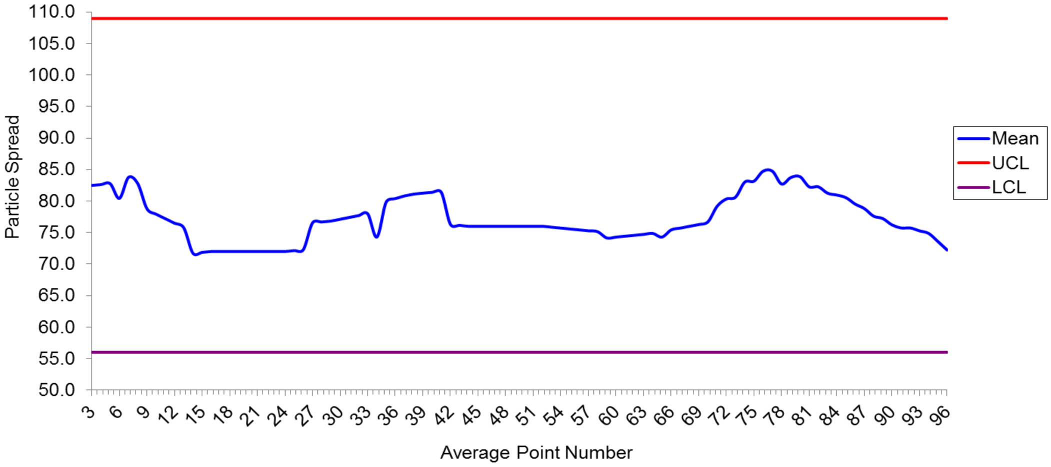


Figure 4.48 Shewhart control chart analysing particle spread computed from the PBI medial axis for a tennis ball experiencing occlusion.

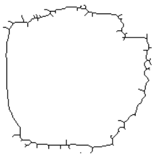


Figure 4.49 Particle spread computed from the PBI medial axis for a table tennis bat experiencing self-occlusion.

As occlusion occurs, the particle spread data and process-behaviour charts show that:

- The PBI medial axis allows measurements to be made in regions of the PBI which are shrunk as a result of occlusion, as shown clearly in Figure 4.41, Figure 4.43 and Figure 4.45. Similar measurement could not be done when using tracker's path estimate. However, since the shrunk region are not clearly visible in the PBI (Figure 4.47) due to the particles becoming stationary once occlusion has occurred. Therefore, no measurement were made as a result of no medial axis being present at this region.

- Shewhart control charts in Figure 4.42 and Figure 4.46 both correctly detect the event of occlusion. Although the control chart mean in Figure 4.46 exceeds the upper control limit momentarily, the exceeding period is too short to cause the tracker firing prematurely. However, control chart results in Figure 4.44 and Figure 4.48 record false positives. This occurrence is explained in section 4.5.3.6.
- In the event of self-occlusion (Figure 4.49), no medial axis path was generated. This is because particle sets generated at later time steps tend to overlap particle sets from prior time steps resulting in a blob like PHI and PBI. The PBI is shown in Figure 4.5 (e). Particle sets that become closely clustered as a result of self-occlusion are not distinguishable in the generated PHI and PBI. Applying pre-processing approaches to the PBI generated a medial axis point instead of a medial axis path. No process-behaviour chart is shown here.

4.5.3.5. Camouflage

Figure 4.50, Figure 4.52, Figure 4.54 and Figure 4.56 show particle spread estimates computed from the medial axis for video sequences showing a target experiencing camouflage. Figure 4.51, Figure 4.53, Figure 4.55 and Figure 4.57 shows corresponding process-behaviour chart graphs.

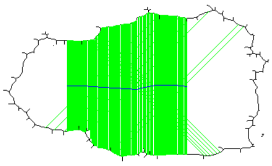


Figure 4.50 Particle spread computed from the PBI medial axis for a yellow circle experiencing camouflage.

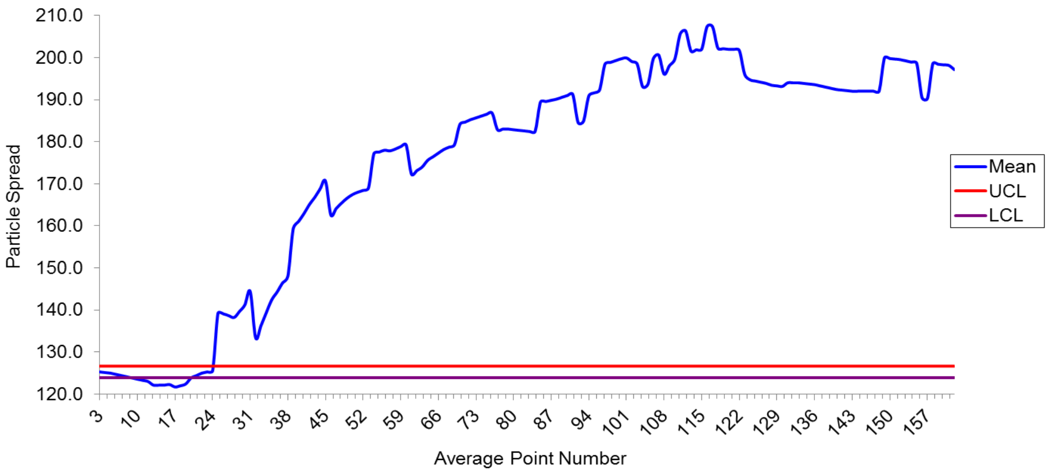


Figure 4.51 Shewhart control chart analysing particle spread computed from the PBI medial axis for a yellow circle experiencing camouflage.

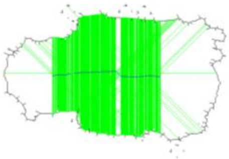


Figure 4.52 Particle spread computed from the PBI medial axis for a yellow circle experiencing camouflage amid the presence of clutter.

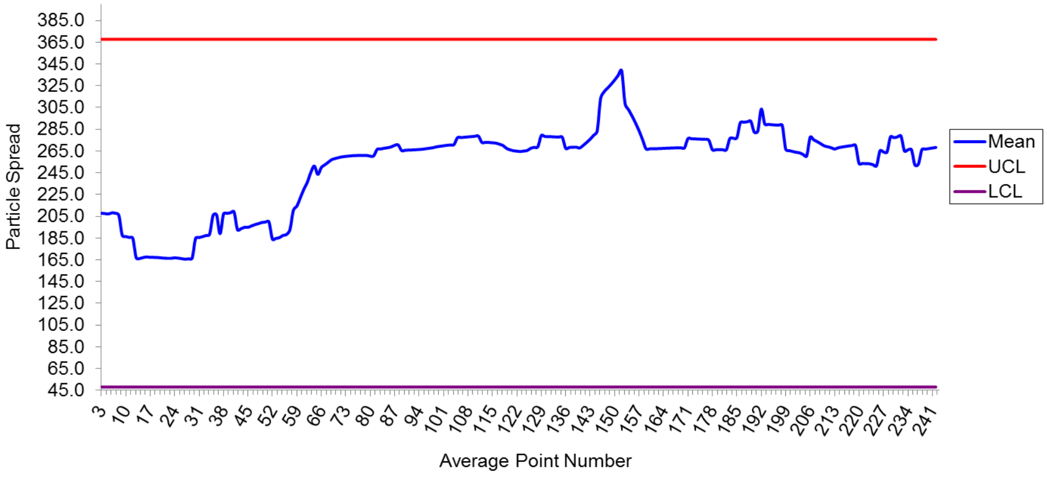


Figure 4.53 Shewhart control chart analysing particle spread computed from the PBI medial axis for a yellow circle experiencing camouflage and with clutter present.

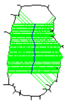


Figure 4.54 Particle spread computed from the PBI medial axis for a player camouflaging a football.

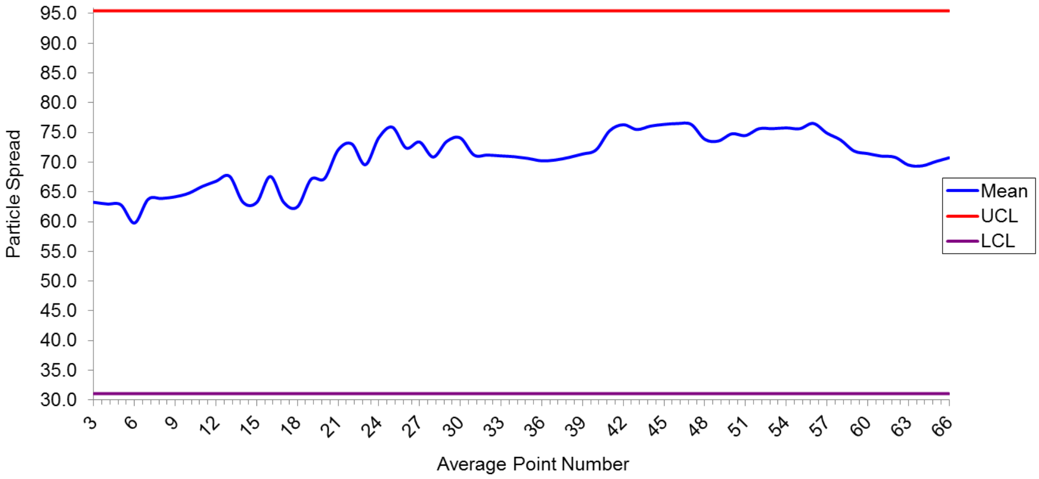


Figure 4.55 Shewhart control chart analysing particle spread computed from the PBI medial axis for a player camouflaging a football.

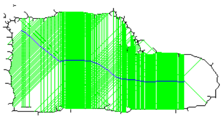


Figure 4.56 Particle spread computed from the PBI medial axis for a tennis ball experiencing camouflage.

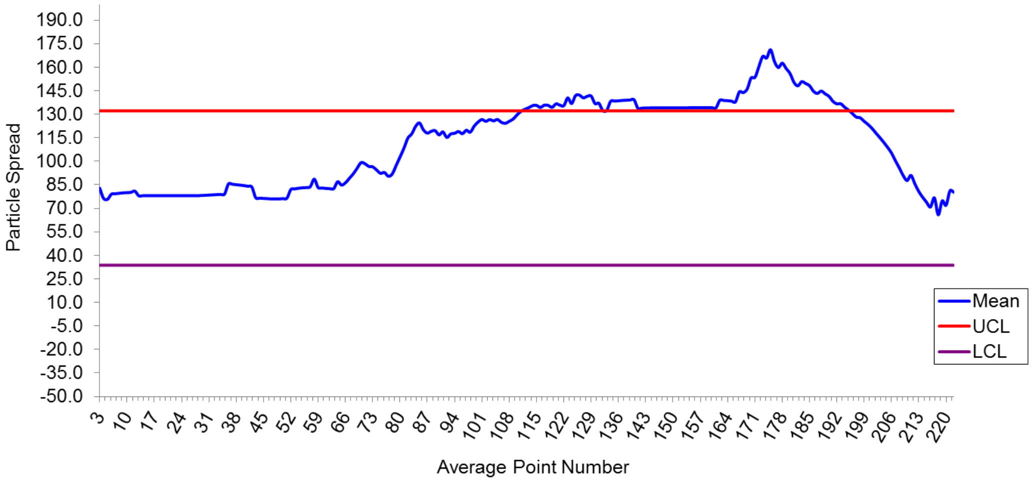


Figure 4.57 Shewhart control chart analysing particle spread computed from the PBI medial axis for a tennis ball experiencing camouflage.

Examination of the particle spread data and process-behaviour charts show that in the event of camouflage:

- The medial axes in Figure 4.50, Figure 4.52, Figure 4.54 and Figure 4.56 allow measurements to be made in the expanded region of the PBI where camouflage occurs, providing more accurate event detection than seen when using the tracker's path estimate.
- Shewhart control charts in Figure 4.51 and Figure 4.57 shows the control mean exceeding the upper control limit in the event of camouflage. However, Figure 4.53 and Figure 4.55 record false positives. This is explained in section 4.5.3.6.

4.5.3.6. Summary

Examination of the particle spread data and process-behaviour charts presented here leads to the following summary:

- Results presented in section 4.5.2 shows that when tracker's path estimate experiences noise due to disruptive events, measurement of the shrunk/expanded regions of the PBI as a result of these events can not be made. This results in a lower quality measurement of the particle spread being made. In contrast, the PBI medial axis does not experience noise when these events occur, thus, allowing measurement of the shrunk/expanded regions of the PBI to be made. Therefore, the PBI medial axis provides more and better oriented and so higher quality measurements of particle spread than the tracker's path estimate. This improves the accuracy of the event detection.

- The false positives recorded in Figure 4.44, Figure 4.48, Figure 4.53 and Figure 4.55 occurs because though the medial axes are typically smooth, the boundary descriptions extracted via texture edge detection may not be. As a result, the control limits computed during the early stages of tracking can be inappropriate, and generate spurious events.
- The PBI medial axis method recorded four false positives as compared to six false positives recorded for the tracker’s path estimate method in section 4.5.2. Tracker’s path estimate method recorded five true positives of detecting the events correctly. However, the PBI medial axis method recorded six true positives of correctly detecting the events, even though self-occlusion could not be measured using the PBI medial axis method.

Analysis shows that the medial axis provides a better representation of the PBI than the tracker’s path estimate. The medial axis is smoother, making the approach more viable than the method presented in section 4.5.2. However, due to noise in the PBI boundaries, the associated produced process-behaviour chart(s) record false positives. This could be addressed by forcing the boundary to become smooth, for example by taking an active contour approach. Alternatively, an alternative, control chart could be adopted.

4.5.4. Building control charts from training sets

All event detection results presented hitherto in this thesis rely upon the process-behaviour chart detailed in Chapter 3. The control limits for this method are computed independently, from a few frames at the beginning of each sequence, when tracking commences. This is the intended use of the Shewhart method. In the method presented here, control limits are only computed after analysing the entire duration of a training sequence in which the target of interest is clearly visible. The lengths of these training sequences are as follows: (1) a clearly visible yellow circle is 196 frames; (2) a yellow circle, clearly visible but in the presence of clutter, is 172 frames and (3) a clearly visible tennis ball is 99 frames. It should be pointed out here that the method of computing and utilising the control limits are similar to the originals, presented in Chapter 3. They only differ in the duration of time it takes to compute them. These control limits are then applied to similar sequences in which occlusion or camouflage might occur. The moving window size is maintained at 7 throughout.

Training data is unavailable for some of the test sequences used previously: (1) the table tennis bat experiencing self-occlusion, (2) the tree occluding a woman and (3) the player camouflaging a football. These examples are therefore not considered further.

The process-behaviour chart graphs given in Figure 4.58, Figure 4.59 and Figure 4.60 show data obtained from a yellow circle tracked while (1) moving normally, (2) experiencing occlusion and (3) experiencing camouflage.

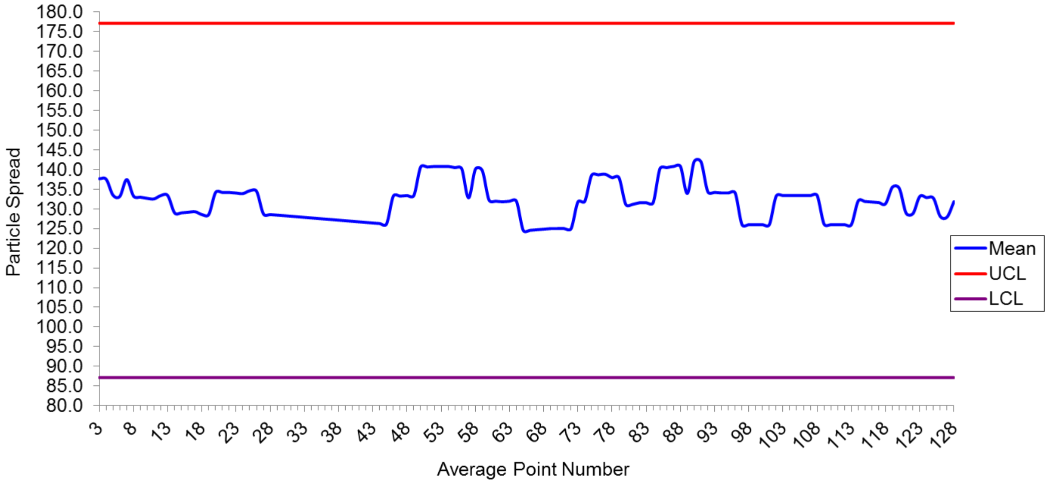


Figure 4.58 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a yellow circle moving normally.

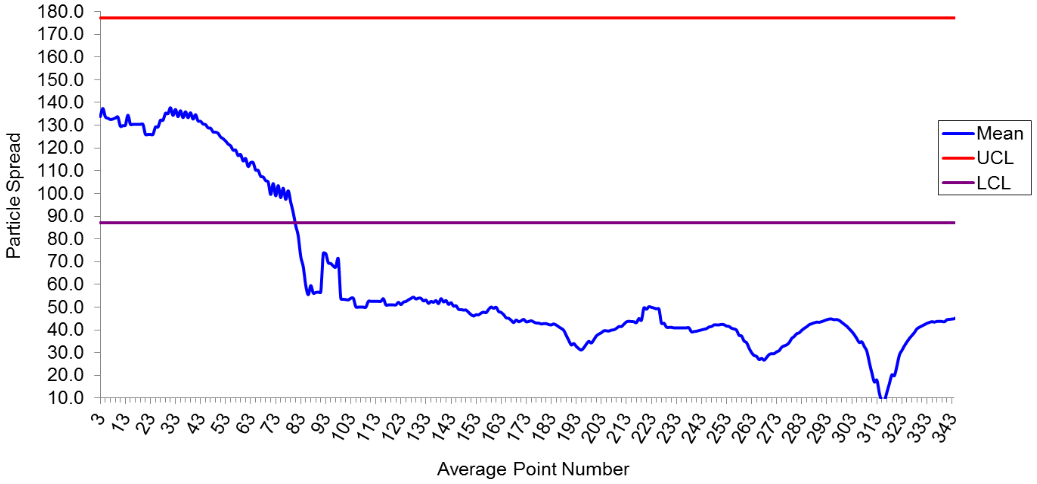


Figure 4.59 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a yellow circle experiencing occlusion.

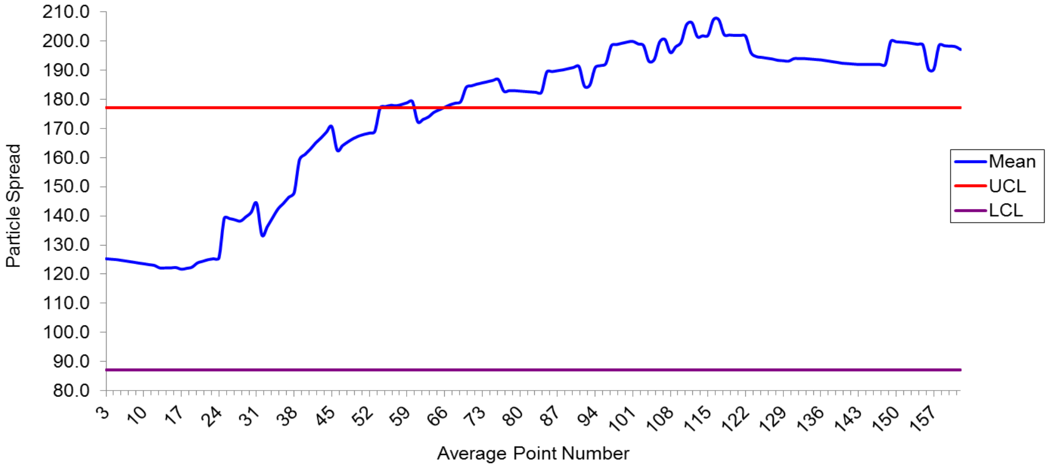


Figure 4.60 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a yellow circle experiencing camouflage.

Figure 4.61, Figure 4.62 and Figure 4.63 show process-behaviour chart graphs produced from videos showing a yellow circle moving through clutter: (1) normally, (2) during occlusion and (3) during camouflage.

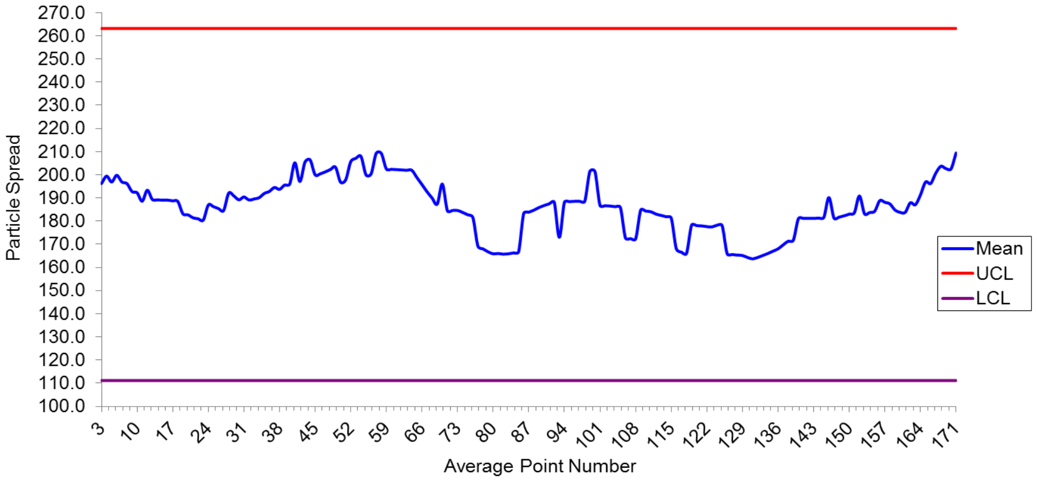


Figure 4.61 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a yellow circle moving normally in the presence of clutter.

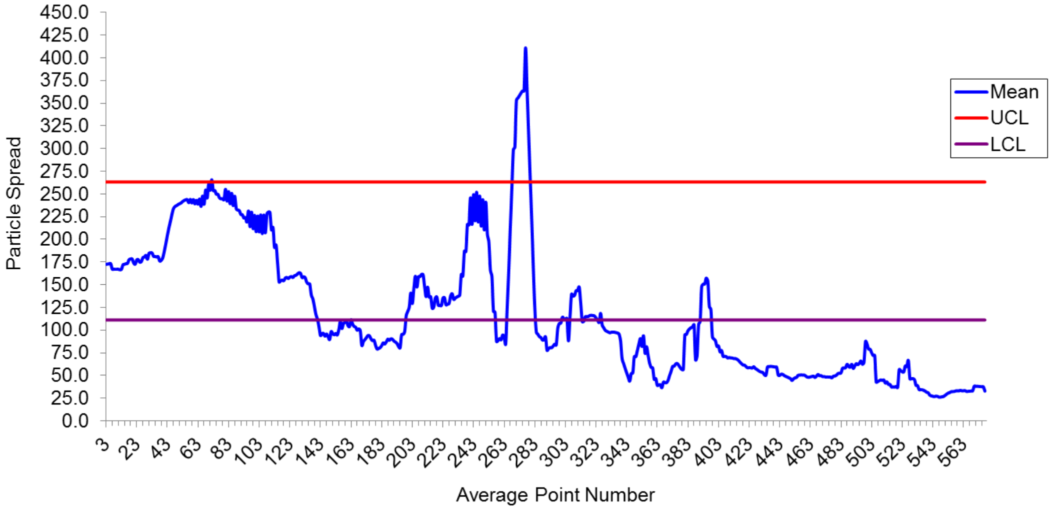


Figure 4.62 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a yellow circle experiencing occlusion and with clutter present.

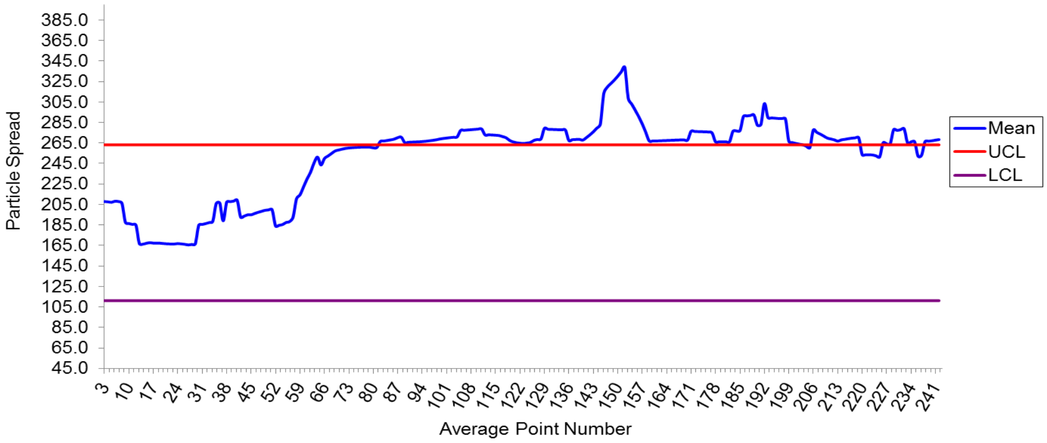


Figure 4.63 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a yellow circle experiencing camouflage in the presence of clutter.

Process-behaviour chart graphs produced from sequences showing a tennis ball being tracked while: (1) moving normally, (2) experiencing occlusion and (3) experiencing camouflage are shown in Figure 4.64, Figure 4.65 and Figure 4.66.

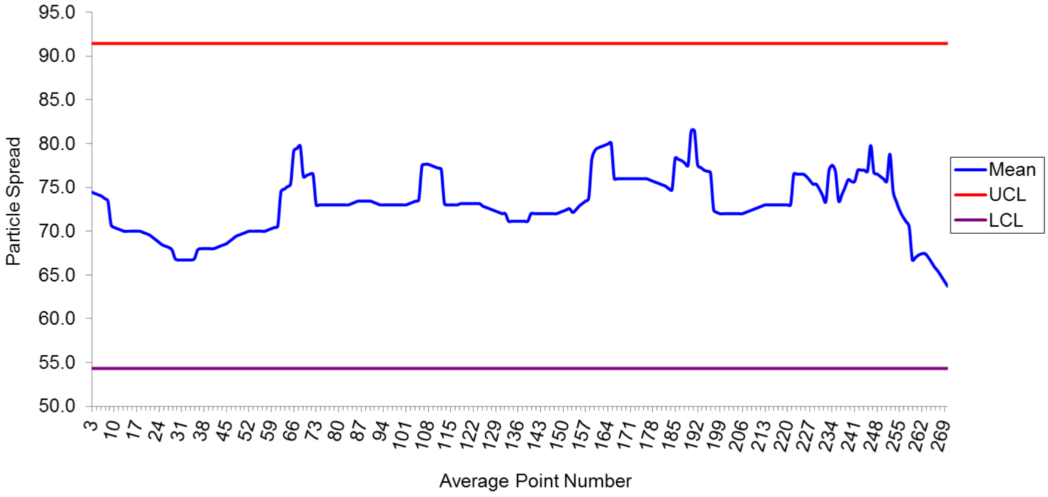


Figure 4.64 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a tennis ball moving normally.

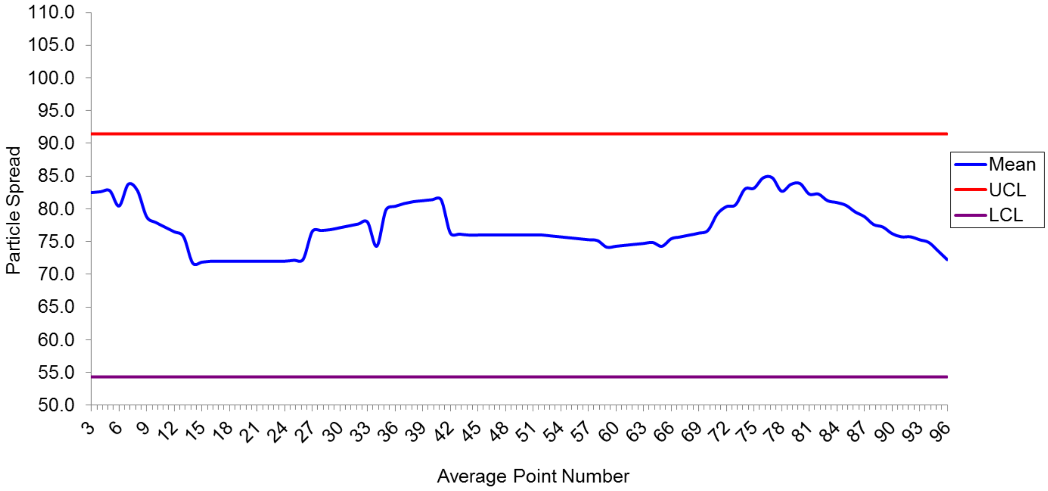


Figure 4.65 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a tennis ball experiencing occlusion.

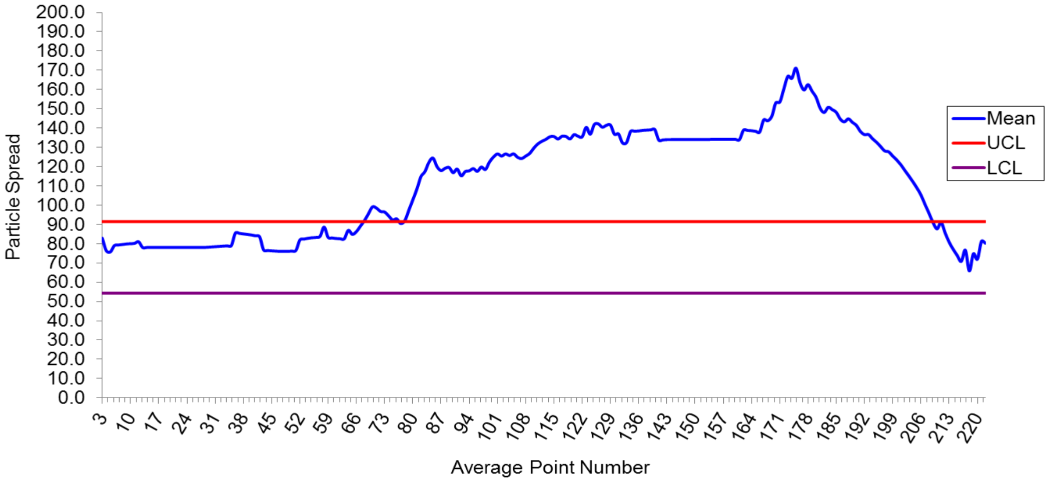


Figure 4.66 Shewhart control chart analysing particle spread computed from the PBI medial axis and via a training set for a tennis ball experiencing camouflage.

Examination of the data presented above shows that:

- Computing control limits from a longer training sequence does improve the accuracy of the process-behaviour chart. All but one of the process-behaviour charts records true positives.
- The process-behaviour chart produced for the sequence showing a tennis ball experiencing occlusion, however, records a false positive (refer to Figure 4.65). This occurs due to the medial axis being within regions of the PBI (refer to Figure 4.47) which had not fully shrunk as a result of occlusion. These regions have similar characteristics as to when a tennis ball is being tracked moving normally. Additionally, the medial axis path does not extend into the shrunk regions of the PBI. Again, it should be noted here that the shrunk regions of the PBI are not clear in Figure 4.47 is because the particles stopped moving once occlusion had occurred. A clearer shrunk region of the PBI would contribute to the medial axis path prolonging into these regions.

4.5.5. Summary

In summary:

1. The tracker's path estimate and the medial axis both provide medial representations of the PBI, with the medial axis being the smoothest and so best suited to the detection of occlusion and camouflage.
2. Computing a template of the process-behaviour chart control limits from a training set improves the accuracy of the process-behaviour chart when detecting occlusion and camouflage.

Drawing comparisons between the medial axis and training set detection method presented here, and the particle clustering method presented in Chapter 3, show that:

- The medial axis and training set method provides opportunities to measure particle spread at a given time point while taking into account particle behaviour over a longer time period. The generated PBI also provides a clearer view of the changes taking place within the configuration of the particle spread as a result of occlusion and camouflage. These are not possible using the particle clustering method in Chapter 3 as the method relies upon particle clusters formed, independently, at each time step. Therefore, the detection method presented here allows a more reliable and consistent evaluation to be made.

- As mentioned earlier, control limits generated for the particle clustering method in Chapter 3 are computed by only analysing data in a few frames at the beginning of a video sequence. Though, results showed that the process-behaviour chart correctly distinguish the events that occur, however, the charts can be sensitive to false data and fire prematurely. Using training sets as shown here allows control limits to be computed by analysing through the entire duration of a video sequence showing a target moving normally. This not only improves the accuracy of the process-behaviour chart but also makes the process-behaviour chart less sensitive and prevents it from firing prematurely.

4.6. A path alignment approach

4.6.1. Motivation

If a clearly visible target is tracked successfully throughout an image sequence, the corresponding PHI will show a broad swath of particles centred on the target path. The width of this particle-filled area will be approximately constant. As a result, distance between the medial axis of the PHI and the estimated target position at each time step will be small. We hypothesise that by measuring the alignment distance between these two paths will provide valuable information to indicate the occurrence of occlusion and camouflage.

4.6.2. Aim

The aims of this section are to:

1. Gather information pertaining to the alignment of the tracker’s path estimate over the medial axis path.
2. Exploiting the gathered information using a process-behaviour chart to detect the occurrences of occlusion and/or camouflage during tracking by monitoring the process-behaviour chart control points.

4.6.3. Alignment analysis

The alignment computation between the tracker’s path estimate and the medial axis path is analysed in viewpoint from the tracker’s path estimate.

Analysis results presented in section 4.5.4 shows that using training sets produces more accurate process-behaviour chart results. As a result, the same approach is applied here.

4.6.3.1. Implementation

Prior to computing the alignment distance between each mean particle on the tracker's path estimate from the medial axis path, a Gaussian smoothing is applied onto the tracker's path estimate. Implementing a Gaussian smoothing has already been detailed in section 4.5.2. Subsequently, a normal line for each mean particle on the tracker's path estimate is computed in the same manner as detailed in section 4.5.2.

Once a normal line is computed, navigate along the line in both directions originating from $(\mu_{x_{t-1}}, \mu_{y_{t-1}})$ until a point (x_{mp}, y_{mp}) on the medial axis path is reached. Then compute the alignment distance between $(\mu_{x_{t-1}}, \mu_{y_{t-1}})$ and (x_{mp}, y_{mp}) using (4.22):

$$d(x,y) = \sqrt{\left(\mu_{x_{t-1}} - x_{mp}\right)^2 + \left(\mu_{y_{t-1}} - y_{mp}\right)^2} \tag{4.22}$$

The direction which has the lowest alignment distance is recorded and later exploited using the process-behaviour chart. However, if no point (x_{mp}, y_{mp}) on the medial axis path is reached, no alignment distance value is computed for that particular mean particle and that mean particle is ignored during the process-behaviour chart analysis. In the graphical representations, normal lines are shown as green lines. Blue dots represent the mean particle on the tracker's path estimate while black lines represent the medial axis path.

Once all the distance results are computed, the alignment distances are then exploited using the process-behaviour chart to detect the occurrences of occlusion and/or camouflage. The same parameter settings applied when computing the particle spread based on temporal behaviour is maintained throughout.

4.6.3.2. Normal tracking

Figure 4.67, Figure 4.69 and Figure 4.71 shows alignment distance for a target moving normally. The associated process-behaviour charts are shown in Figure 4.68, Figure 4.70 and Figure 4.72.

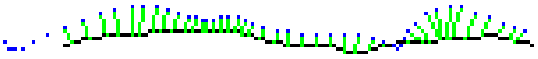


Figure 4.67 Alignment distance for a yellow circle moving normally.

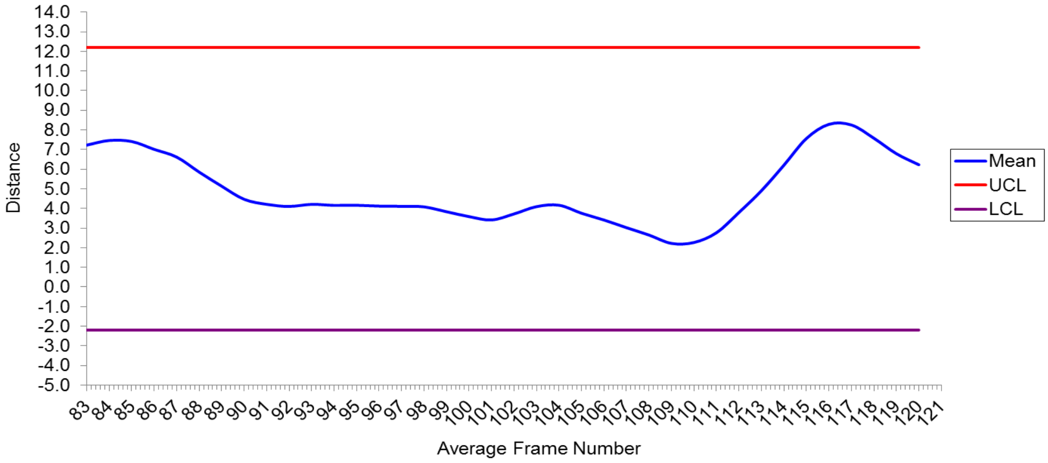


Figure 4.68 Shewhart control chart analysing the alignment distance between the paths for a yellow circle moving normally.

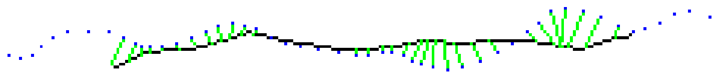


Figure 4.69 Alignment distance for a yellow circle moving normally amid the presence of clutter.

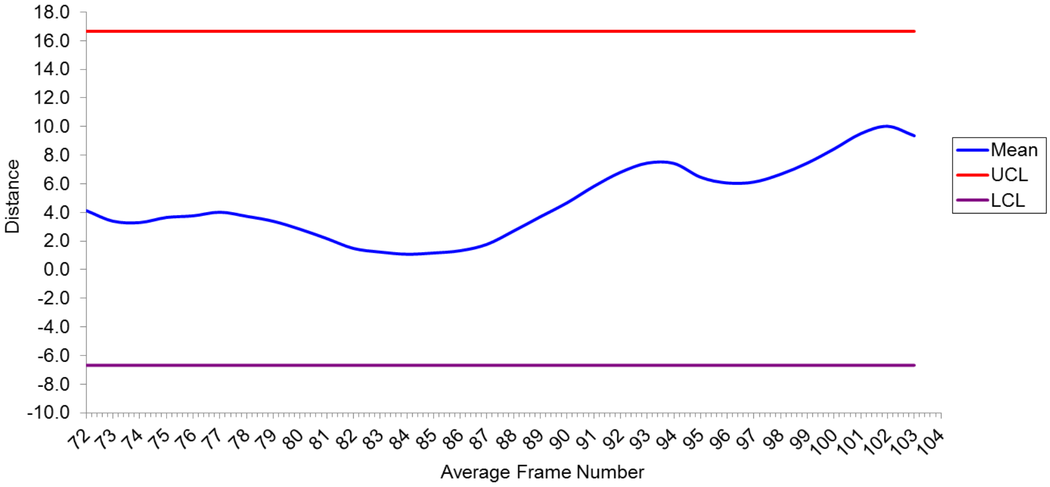


Figure 4.70 Shewhart control chart analysing the alignment distance between the paths for a yellow circle moving normally amid the presence of clutter.

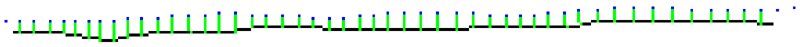


Figure 4.71 Alignment distance for a tennis ball moving normally.

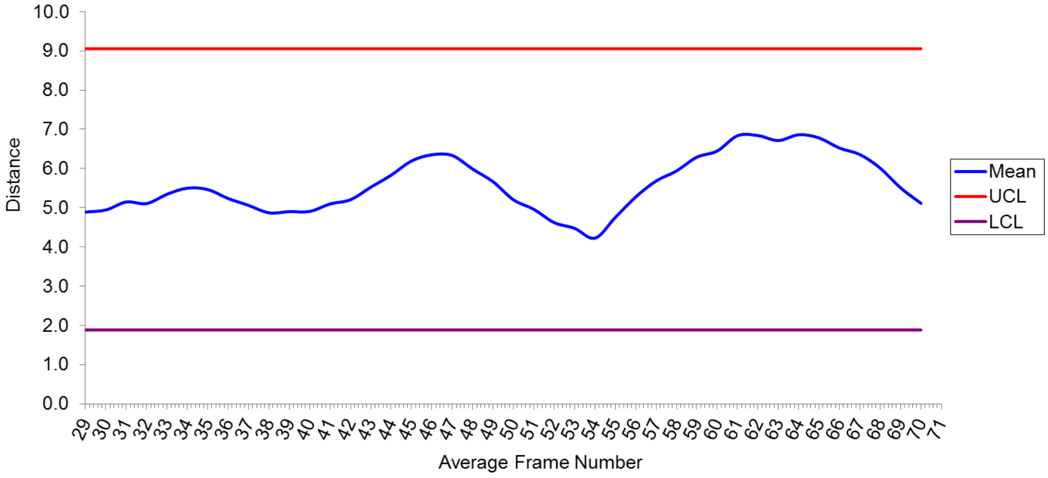


Figure 4.72 Shewhart control chart analysing the alignment distance between paths for a tennis ball moving normally.

During normal tracking:

- Figure 4.67, Figure 4.69 and Figure 4.71 all show the alignment between the tracker's path estimate and the medial axis path to be in close proximity throughout the video sequence.
- Since the alignment distance between the two paths remain relatively constant throughout, results in the control mean in the Shewhart control charts (Figure 4.68, Figure 4.70 and Figure 4.72) remained within the boundaries of the control limit for the duration of the video sequence. This also contributed in all three charts recording true positives.

4.6.3.3. Occlusion

Alignment distance when a target experiences occlusion is shown in Figure 4.73, Figure 4.75 and Figure 4.77. Figure 4.74, Figure 4.76 and Figure 4.78 shows the process-behaviour charts produced for these videos sequence.

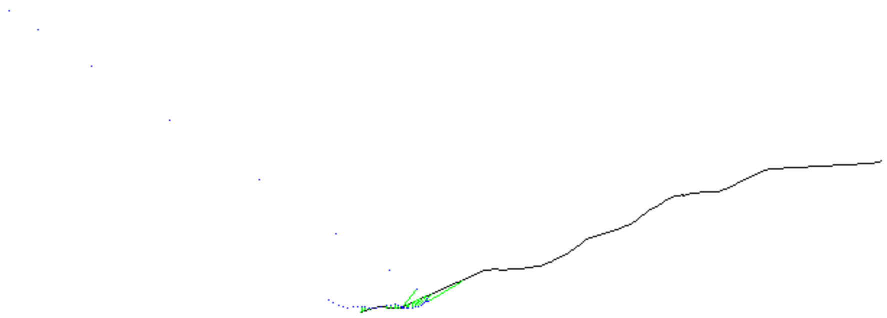


Figure 4.73 Alignment distance for a yellow circle experiencing occlusion.

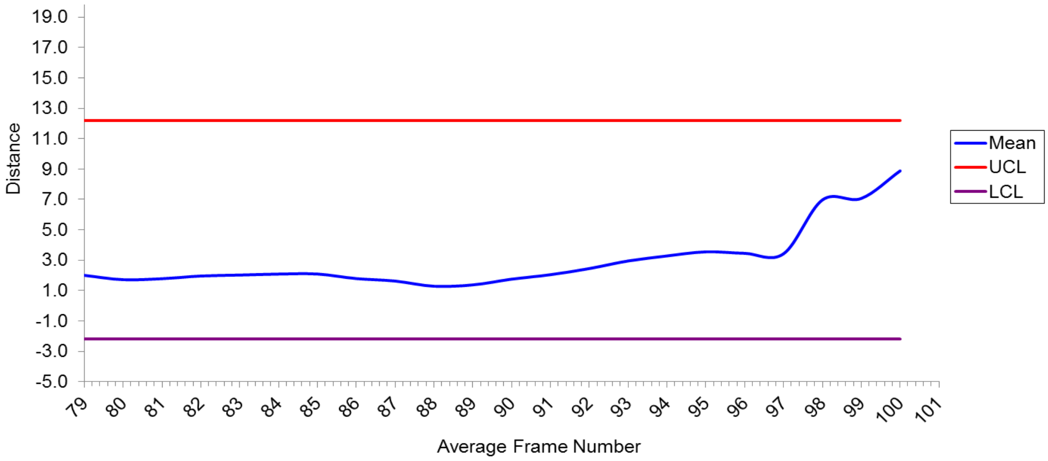


Figure 4.74 Shewhart control chart analysing the alignment distance between the paths for a yellow circle experiencing occlusion.

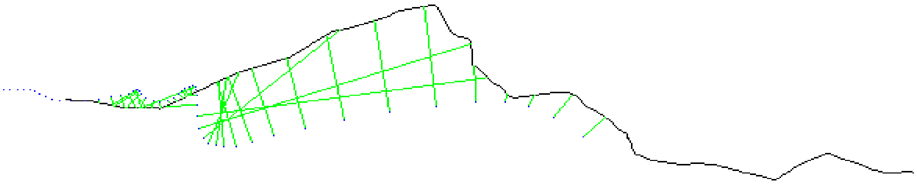


Figure 4.75 Alignment distance for a yellow circle experiencing occlusion amid the presence of clutter.

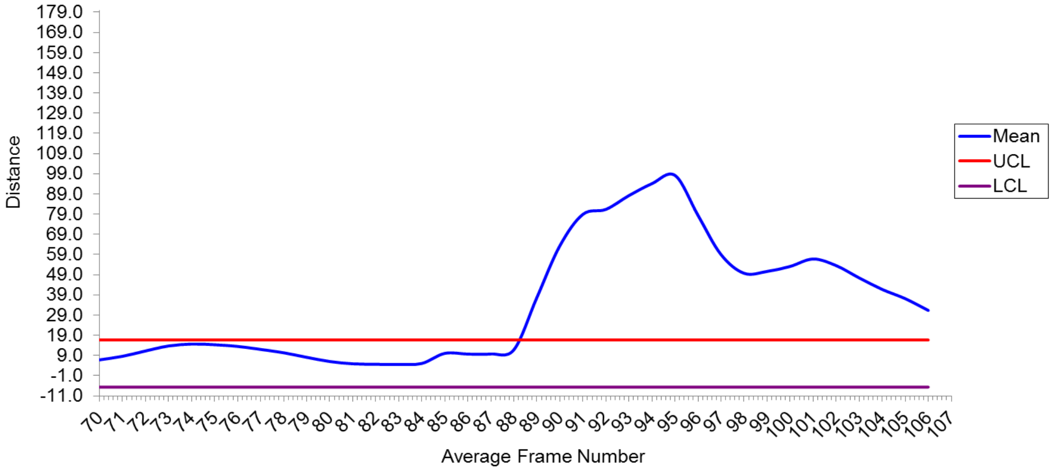


Figure 4.76 Shewhart control chart analysing the alignment distance between the paths for a yellow circle experiencing occlusion amid the presence of clutter.

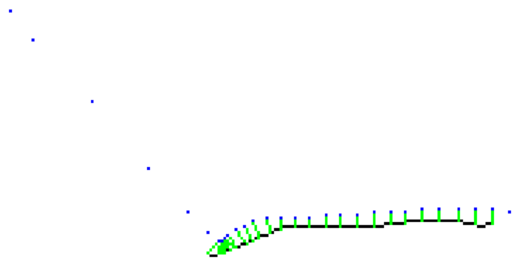


Figure 4.77 Alignment distance for a tennis ball experiencing occlusion.

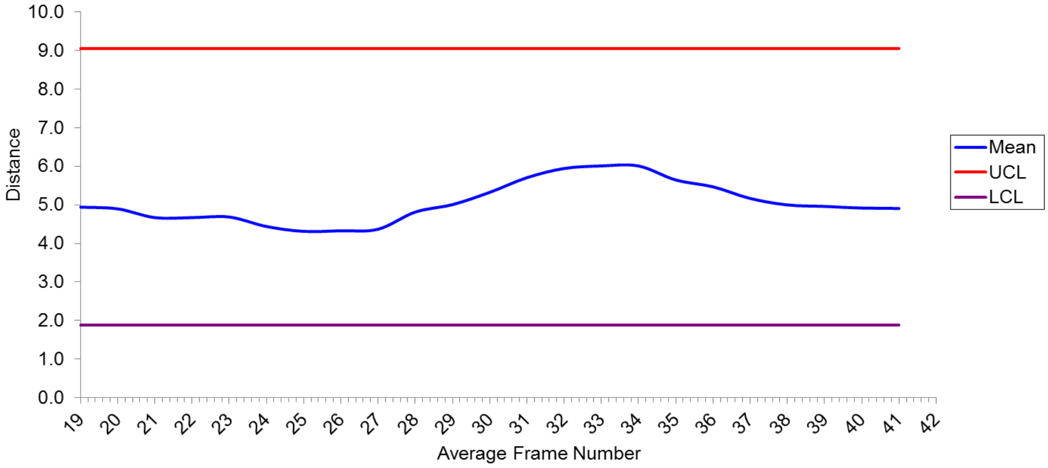


Figure 4.78 Shewhart control chart analysing the alignment distance between the paths for a tennis ball experiencing occlusion.

In the event of occlusion:

- The path alignment between the paths remains in close proximity while the target is fully visible. However, when occlusion happens, Figure 4.73, Figure 4.75 and Figure 4.77 all show the tracker's path estimate deviating away from the medial axis path due to the tightly clustered particle set dispersing as a result of occlusion. This causes the alignment distance between the paths to increase.
- Shewhart control chart in Figure 4.76 shows the control mean exceeding the upper control limit when occlusion is detected. Results presented thus far have shown that the occurrence of occlusion causes the mean to exceed the lower control limit. This is not seen here because a transitory increase in alignment distance is seen during occlusion. This will only result in an increase in the control mean, which implies that the mean will never trigger the lower limit of the chart. Figure 4.74 and Figure 4.78 both recorded false positives. The control mean in both of these charts

remained within the control limit boundaries even after occlusion has taken place. The reason for both of these charts recording false positives will be explained in section 4.6.3.5.

4.6.3.4. Camouflage

Graphical representations of alignment distance when a target experiences camouflage are shown in Figure 4.79, Figure 4.81 and Figure 4.83. The associated process-behaviour charts graphs are shown in Figure 4.80, Figure 4.82 and Figure 4.84.

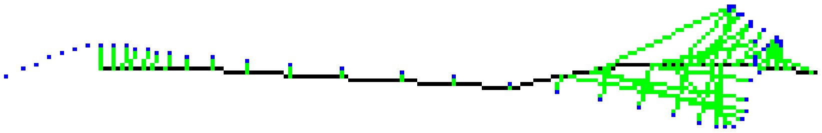


Figure 4.79 Alignment distance a yellow circle experiencing camouflage.

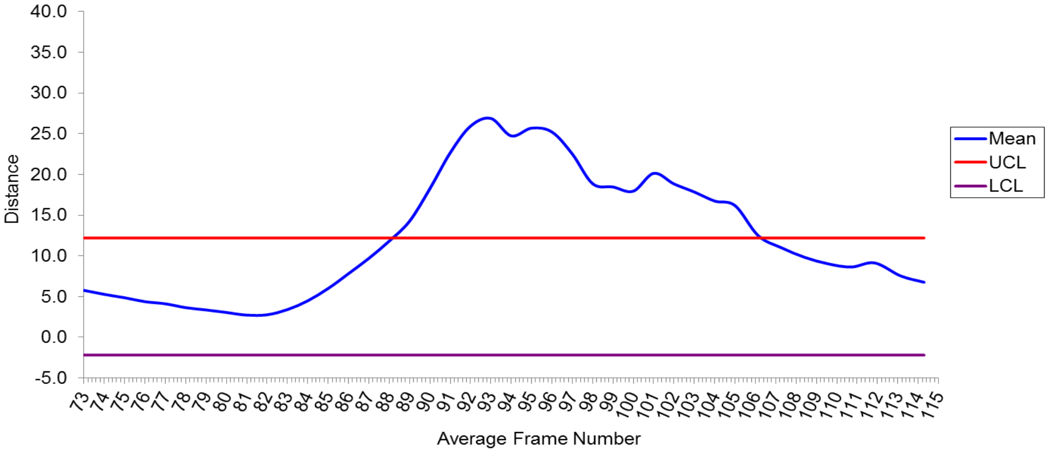


Figure 4.80 Shewhart control chart analysing the alignment distance between the paths for a yellow circle experiencing camouflage.

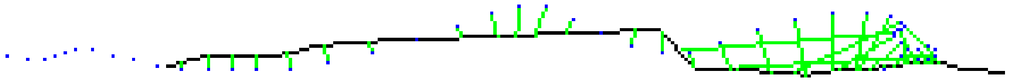


Figure 4.81 Alignment distance for a yellow circle experiencing camouflage amid the presence of clutter.

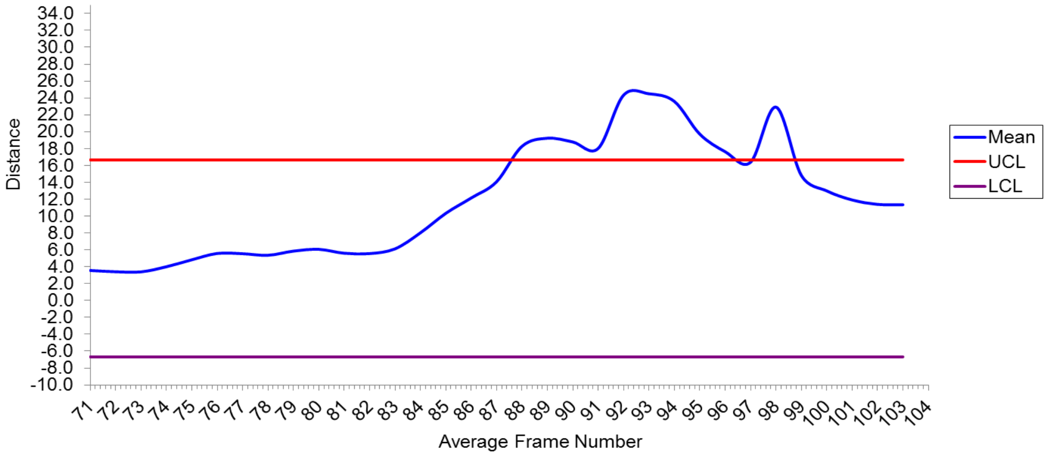


Figure 4.82 Shewhart control chart analysing the alignment distance between the paths for a yellow circle experiencing camouflage amid the presence of clutter.

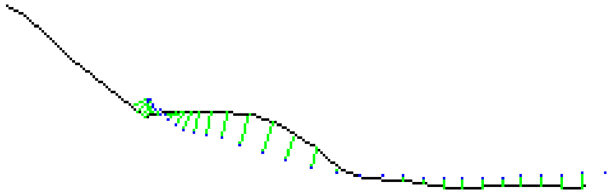


Figure 4.83 Alignment distance for a tennis ball experiencing camouflage.

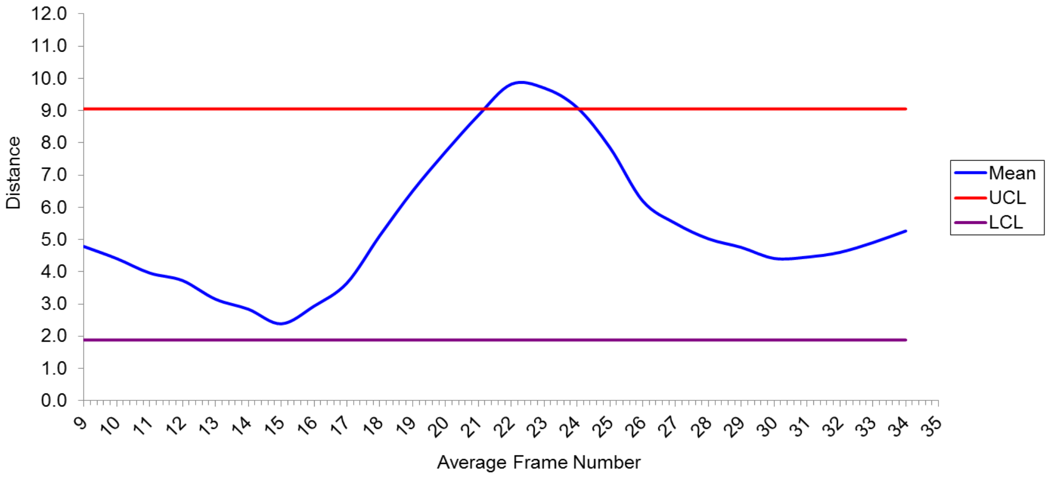


Figure 4.84 Shewhart control chart analysing the alignment distance between the paths for a tennis ball experiencing camouflage.

In the event of camouflage:

- As in the occlusion analysis, the path alignment between the tracker's path estimate and the medial axis path in Figure 4.79, Figure 4.81 and Figure 4.83 all show both paths being in close proximity when tracking the target moving normally. However, the alignment distances all experiences a sudden increase as camouflage occurs. This is because as camouflage occurs, the PHI will begin to reflect the shape of the camouflaging object and the tracker's path estimate can reasonably be expected to deviate away from the medial axis.

- The increase in the alignment distance causes the control mean in the Shewhart control chart in Figure 4.80, Figure 4.82 and Figure 4.84 to exceed the upper limits of the chart. It should be pointed out here that all three charts record true positives of the event.

4.6.3.5. Summary

Examination of the alignment distance leads to the following summary:

- Occlusion and camouflage can be detected by estimating errors in the alignment of the medial axis of the PHI and the tracker's path estimate. This presents an alternative approach to detecting these disruptive events as compared to the particle clustering method in Chapter 3 and the medial axis and training set method in section 4.5.4, which both detect occlusion and camouflage by detecting anomalies in the particle spread via process-behaviour chart. Even though, results shows that evaluating the alignment distance data via process-behaviour chart respond differently to occlusion and camouflage as well as cannot clearly distinguish these disruptive events apart, nevertheless, results also shows that evaluation of the alignment distance via process-behaviour chart is still effective at detecting these events.
- The alignment distance method recorded seven true positives when detecting disruptive events.
- Only two events recorded false positives (refer to Figure 4.74 and Figure 4.78). This is because:

    1. The result shown in Figure 4.74 measures the result in Figure 4.73. Figure 4.73 shows the tracker's path estimate being in close proximity to the medial axis path as the target is tracked moving normally. However, as occlusion occurs, the tracker's path estimate starts to deviate away and in the opposite direction as to the medial axis path. This occurs because after occlusion, the clustered particle sets starts experiencing fluctuations in its behaviour, thus providing unreliable data. Thus, making it difficult to measure the alignment distance between the paths.

2. The result in Figure 4.78 showed that the use of a moving window size of 7 was too large to detect occlusion correctly. Figure 4.85 shows the mean data going out of control as a result of occlusion when using a moving window size of 3. Using a smaller moving window size makes the process-behaviour chart more sensitive to immediate changes in the mean data as the amount of alignment distance data analysed in one instance is less.

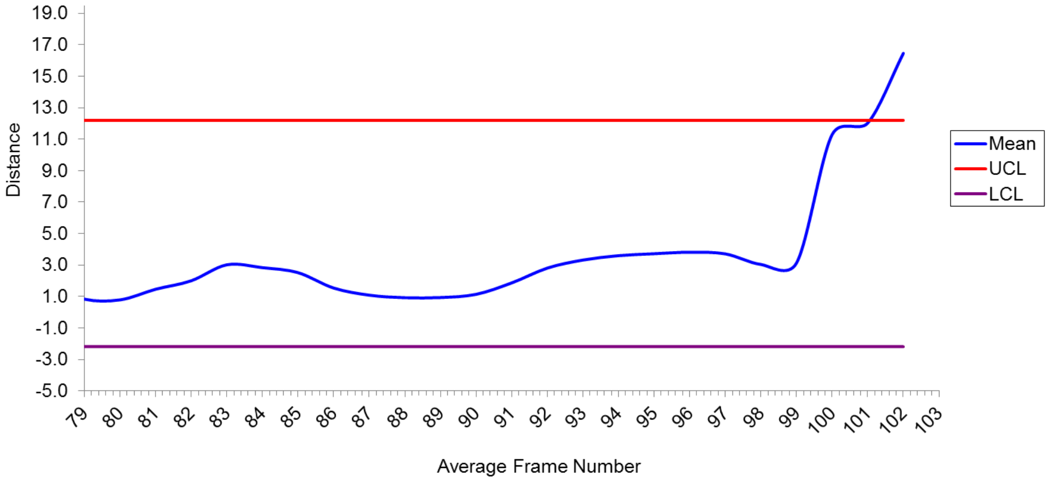


Figure 4.85 Shewhart control chart analysing the alignment distance between the paths for a yellow circle experiencing occlusion using a moving window size of 3.

Although the alignment path method cannot clearly distinguish the events of occlusion and camouflage as done by the particle clustering detection method in Chapter 3 and the medial axis and training set detection method in this chapter, analysis showed the alignment path method to be a viable detection method in detecting disruptive events.

4.7. Chapter summary

Motion analysis and action recognition require compact but rich representations of the spatial configuration and movement of the object(s) of interest over some time period. Inspired by the motion history image (MHI) this chapter has introduced a more general view-based representation, the particle history image (PHI).

The particle history image is constructed not from silhouettes, but from the particle sets employed by particle filter-based tracking algorithms. As such it is independent of shape, making the detection of disruptive events much easier. Examination has shown that creating a PHI over a suitable temporal window presents a chronological view-based representation of the change in particle set behaviour which can be exploited to detect occlusion and camouflage events. To achieve this, the boundary of the particle mass was obtained using texture edge detection, producing a secondary representation, the particle boundary image (PBI).

Building upon the results obtained in Chapter 3, two measurements of the particle distribution were presented:

1. Examination has shown that measurement made emphasising on spatial information in the PHI and PBI provides a more viable and robust measure of the particle spread than using temporal information in the PHI and PBI. Analysis also provided evidence that using PHI and PBI spatial information together with a template of the process-behaviour chart control limits computed from training sets improves the accuracy of occlusion and camouflage detection.
2. Investigation also showed that the detection of occlusion and camouflage can be made by exploiting the alignment distance of the medial axis of the PHI and the tracker's path estimate. Although, the detection method cannot clearly distinguish the disruptive events that occur, but it does provide an accurate indication when the target is no longer moving normally.

The two measurements of the particle distribution presented in this chapter are evaluated in Chapter 5 to detect occlusion and/or camouflage when tracking multiple targets in a static scene.

Chapter 5 Scene Mapping

5.1. Aim

Results presented in the previous two chapters showed that the detection of occlusion and camouflage can be accomplished using process-behaviour charts to exploit data obtained from one of three approaches: (1) instantaneous particle clustering; (2) measurement of particle spread from the medial axis of the PHI; and (3) estimation of errors in the alignment of the PHI's medial axis and the tracker's estimate of target path. Henceforth, these three approaches will be referred to by the numbers (in brackets) above.

As a result, the aims of this chapter are to:

1. Evaluate the effectiveness of each approach at detecting occurrences of disruptive events when tracking multiple targets moving within a static scene.
2. Exploit the resulting information to:

   a) Build scene maps showing the boundaries of interfering objects, providing valuable information about the viewed environment.

   b) Assess the accuracy of these scene maps.

5.2. Motivation

A scene map provides valuable information about the environment being viewed. The process of creating a scene map is commonly called scene mapping. In order to produce a scene map, data acquired at different time points must be fused and the combined data presented in an appropriate, spatially indexed format. Scene maps are often registered with the ground plane, with objects etc., locations being projected onto that plane. The alternative, adopted here, is to project object and/or event data onto a plane registered with the imaging surface of a camera viewing the scene. Scene maps are important in fields such as augmented reality (Castle et al., 2011a, Castle et al., 2011b, Ventura et al., 2012) and robotics (Ess et al., 2009) where maps are fundamentally used for obstacle avoidance (Morris et al., 2008), path planning (Dolgov et al., 2008) and robot navigation (Holz et al., 2008).

In what follows, multiple targets are tracked while moving within a static scene. The information gathered is fused together to build a map of the viewed environment. The maps are used to: (1) provide information pertaining to the boundary of the interfering object(s), thus identifying the likely locations of disruptive events; (2) assess the ability of each approach to describe the boundary of interfering object(s) in the viewed environment.

5.3. Scene mapping tools

5.3.1. Test videos

For the scene mapping experiments detailed in section 5.4, three sets of test videos were created and captured. Two sets are artificial while one set contains real videos.

5.3.1.1. Artificial videos

A total of two hundred artificial videos were created to exhibit targets experiencing one of three scenarios: moving normally, experiencing occlusion or experiencing camouflage.

In these artificial videos, coloured circles are randomly placed at the boundaries of an image containing an identically (when considering camouflage) or differently (when considering occlusion) coloured rectangle. Figure 5.1 and Figure 5.2 show graphical representations of these events, created by overlaying selected frames (at instances where occlusion or camouflage is about to be invoked) extracted from selected video sequences.

Henceforth, all artificial videos exhibiting occlusion will be referred to as artificial occlusion scene mapping while all artificial videos exhibiting camouflage will be referred to as artificial camouflage scene mapping.

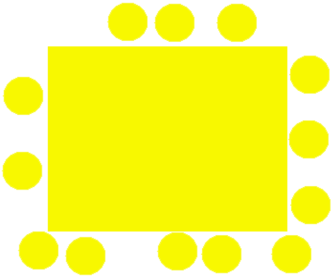


Figure 5.1 A graphical representation of how camouflage is to be invoked.

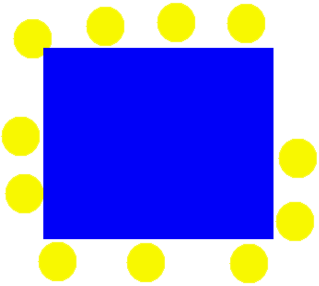


Figure 5.2 A graphical representation of how occlusion is to be invoked.

5.3.1.2. Real videos

One hundred and forty three videos were edited out of an hour long image sequence captured using a fixed camera monitoring an outdoor walkway at the University of Nottingham, Malaysia Campus. The walkway comprises a roof supported by a series of narrow vertical pillars that occlude those using it. Each video shows pedestrians being occluded by the pillars. Figure 5.3 shows a sample image from the video sequence.

Henceforth, all videos exhibiting pedestrian(s) being occluded by the pillars will be referred to as real occlusion scene mapping.

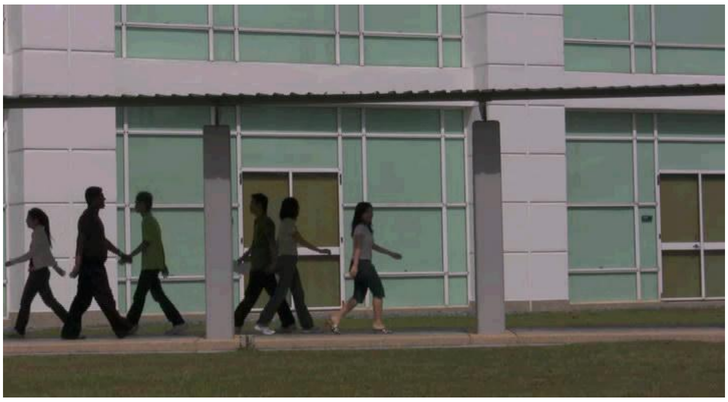


Figure 5.3 An actual scene showing pedestrians being occluded by narrow vertical pillars.

5.3.2. Tracking parameters

5.3.2.1. Artificial videos

For every artificial scene mapping videos analysis, a particle set size of 100 is used. To produce good tracking results for all test videos, a radius size of 3 is used to add process noise. A histogram bin size of 10 is selected for all videos. The same parameter settings are also applied during the creation of the PHIs.

The PBIs are obtained by applying the parameter settings documented in Chapter 4.

5.3.2.2. Real videos

In analysing the real occlusion scene mapping videos, the pedestrians were tracked using Condensation employing an RGB colour histogram target representation. An RGB histogram was chosen after an HSV colour histogram representation produced poor results due to the large number of shadows caused by the strong sunlight. HSV is generally more stable than RGB under illumination changes. This is because illumination changes usually affect the V component much more than the H and S, allowing V values to be down-weighted or ignored. R, G and B are all affected by illumination changes. Here, however, the changes caused by the strong sunlight are very large, and the shadows introduce an additional hue change. Under these circumstances, though each component changes, the combined RGB vector changes more smoothly than the HSV data and therefore was chosen.

For every real occlusion scene mapping videos analysis, a particle set size of 100 was used. Process noise varied depending on the target size. Therefore, a radius size ranging from 1 to 3 was selected, empirically, to produce a good tracking result. A histogram bin size of 10 was used for all videos. The same parameter settings are also used in the creation of the PHIs, and the parameter setting employed in Chapter 4 were again used to obtain PBIs.

5.3.3. Process-behaviour chart control limits

5.3.3.1. Artificial videos

Results in Chapter 4 showed that accuracy of detecting occlusion and camouflage from applying approaches (2) and (3) are significantly improved using a template of the process-behaviour chart control limits, computed from a training set. Thus, the same manner in computing the control limits is also employed here for the artificial scene mapping videos analysis.

The control limits were empirically determined.

5.3.3.2. Real videos

In real scene mapping videos analysis, no video exhibited a pedestrian being tracked moving normally, therefore, providing no training set to compute the control limits. Therefore, the control limits for approach (2) and (3) was computed based on the concept detailed in Chapter 3.

5.3.4. Confusion matrices

Confusion matrices are a visualisation tool commonly utilised when evaluating the performance of a classification system. Each column represents cases in an actual classification while each row represents cases in a predicted classification, table entries show the number of times each predicted classification was made when each actual classification was present. Here, confusion matrices are used to show the accuracy of each approach when differentiating between tracking target(s) moving normally, or experiencing occlusion and/or camouflage. These confusion matrices are presented and discussed in section 5.4.

5.3.5. Implementing occlusion and camouflage map(s)

When a process-behaviour chart fires as a result of some measure exceeding the acceptable range, the associated position (of the particle cluster, or relevant point on the medial axis) and particle spread provide an indication of the image location at which occlusion or camouflage occurred. Note that the position alone may not be reliable, but that the particle spread gives an indication of potential error. If the camera is fixed, results obtained by tracking multiple targets through the same environment can be combined to produce an occlusion or camouflage map of the background scene.

Each time an event is detected, the associated particle spread information is viewed as describing a Gaussian particle distribution and added to a Gaussian Mixture Model representing the occlusion or camouflage structure of the viewed environment. This mixture model is situated on a scene map registered with the image plane. The scene map is initialised to zero, indicating no evidence for an interfering boundary. When a new event is detected, scene map locations are updated with the value of that event's Gaussian, recording the likelihood of an event boundary at that location. When Gaussians overlap, the highest value is recorded. The concept of the Gaussian Mixture Model has already been detailed in Chapter 3 of this thesis. The model used here is somewhat simplified, in that the weight associated with each component is effectively equal.

Though the resulting maps (shown in section 5.4) are used here primarily as indicators of the performance of the event detection approaches they are potentially useful in themselves, providing valuable information about scene structure.

5.4. Detecting and mapping occlusion and camouflage

Table 5.1, Table 5.4 and Table 5.7 present confusion matrix results using approach (1), while, Table 5.2 , Table 5.5 and Table 5.8 are confusion matrix results using approach (2). Confusion matrix results via approach (3) are presented in Table 5.3, Table 5.6 and Table 5.9.

All three video sets are used during the analysis of approach (1). The result is then fused together to generate the respective scene maps. The occlusion and camouflage maps are shown in Figure 5.4 (a), Figure 5.5(a) and Figure 5.6 (a).

However, during the analysis of approaches (2) and (3), only selected videos are used. The chosen videos are randomly selected and comprise videos exhibiting target(s) being tracked moving normally or experiencing either occlusion and/or camouflage. A small number of videos are used because approach (2) and (3) require computation of the PBI. Since the viewed scene over which the analysis is being performed remains constant throughout, using a high number of videos will produce results that are redundant. The occlusion and camouflage maps generated using approach (2) are shown in Figure 5.4(c), Figure 5.5(c) and Figure 5.6(c), while Figure 5.4 (e), Figure 5.5(e) and Figure 5.6(e) give the occlusion and camouflage maps produced via approach (3).

The ability of the proposed methods to describe the occlusion or camouflage structure of the viewed environment is illustrated in Figure 5.4 (b) (d) (f), Figure 5.5 (b) (d) (f) and Figure 5.6 (b) (d) (f). These figures are generated by overlaying the Gaussian Mixture Model results over edge-detected images of the viewed scene where the detected camouflage/occlusion boundaries and the true silhouette of the camouflaging/occluding object are shown in yellow.

Table 5.1 Confusion matrix result for artificial occlusion scene mapping videos using approach (1): particle clustering.

		Actual	
		Normal tracking	*Occlusion*
Predicted	*Normal tracking*	**40**	0
	Occlusion	3	**57**

Table 5.2 Confusion matrix result for artificial occlusion scene mapping videos using approach (2): spread measured relative to the PBIs medial axis.

		Actual	
		Normal tracking	*Occlusion*
Predicted	*Normal tracking*	**4**	0
	Occlusion	1	**10**

Table 5.3 Confusion matrix for artificial occlusion scene mapping videos using approach (3): alignment for the PBI medial axis and tracker's path estimate.

		Actual	
		Normal tracking	*Occlusion*
Predicted	*Normal tracking*	**4**	1
	Occlusion	2	**8**

For artificial occlusion scene mapping, the confusion matrices show that:

- All three approaches detect occlusion but suffer from misclassifications.
- Approach (1) records three misclassifications (Table 5.1), while approach (2) record only one misclassification (Table 5.2). These misclassifications during normal tracking are caused when the target cut across the corner of the interfering occluding object, while still maintaining partial visibility. This instance, however, can cause a change in the particle spread width (as a result of the particles gathering closely together, tracking the partially visible side of the target), hence, causing the detector to fire.
- Approach (3) also records three misclassifications (two during normal tracking and one during occlusion), as shown in Table 5.3. This can result from noise on the path which causes the detector to fire/not fire correctly.

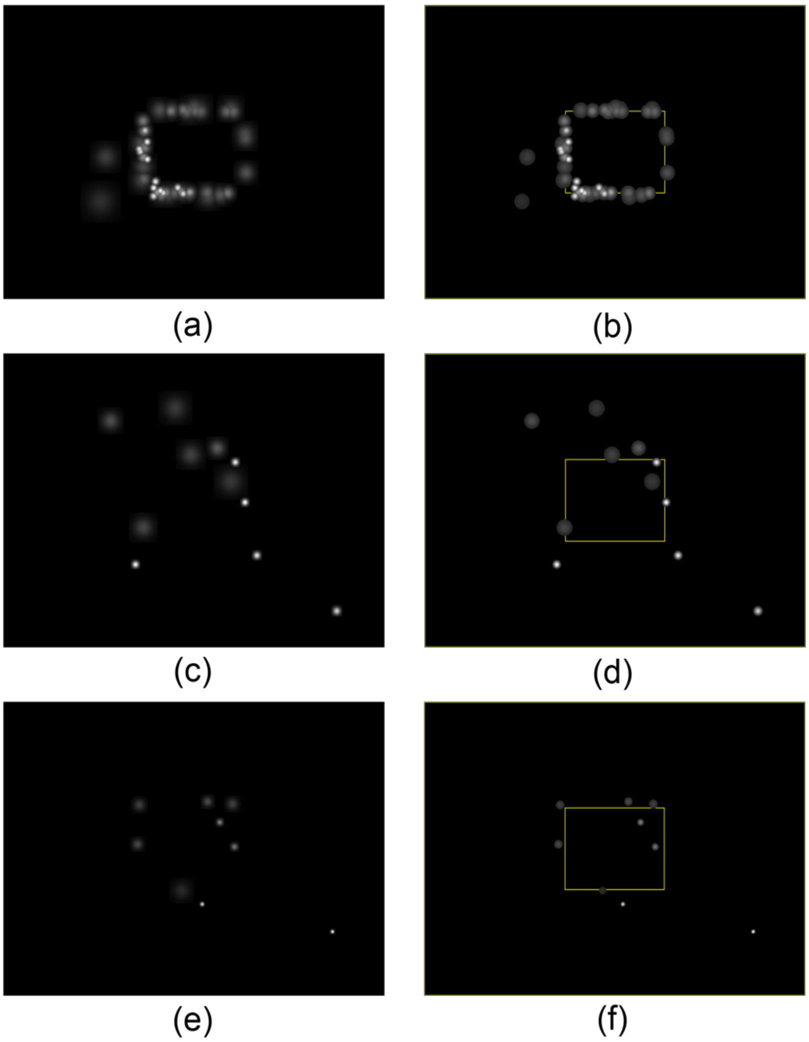


Figure 5.4 Occlusion maps for artificial occlusion scene mapping videos via: (a) approach (1), (c) approach (2), (e) approach (3); (b), (d), and (f) show occlusion maps overlaid on edge-detected images of the viewed scene.

Occlusion maps generated for artificial occlusion scene mapping provide evidence that:

- All three approaches correspond to the true silhouette of the occluding object with the maps (Figure 5.4 (a), (b)) generated via approach (1) providing a slightly clearer illustration with many more data points used.

- All three approaches produce maps (Figure 5.4 (a), (b), (c), (d), (e) and (f)) showing entries gathered outside of the occluding object boundaries. Three entries, however, fire just inside the occluding object (one in Figure 5.4 (d) and two in Figure 5.4 (f)). The single entry fire in Figure 5.4 (d) is due to the target cutting across the corner of the interfering occluding object while still maintaining partial visibility, causing the detector to fire. Whereas, the two entries in Figure 5.4 (f) are due to noise on the path which causes the detector to fire.

Table 5.4 Confusion matrix result for artificial camouflage scene mapping videos using approach (1): particle clustering.

		Actual	
		Normal tracking	*Camouflage*
Predicted	*Normal tracking*	**28**	0
	Camouflage	6	**66**

Table 5.5 Confusion matrix for artificial camouflage scene mapping videos using approach (2): spread measured relative to the PBIs medial axis.

		Actual	
		Normal tracking	*Camouflage*
Predicted	*Normal tracking*	**5**	0
	Camouflage	0	**10**

Table 5.6 Confusion matrix for artificial camouflage scene mapping videos using approach (3): alignment of the PBI medial axis and tracker's path estimate.

		Actual	
		Normal tracking	*Camouflage*
Predicted	*Normal tracking*	**5**	0
	Camouflage	0	**10**

Examining the confusion matrices for artificial camouflage scene mapping shows that:

- Approach (1) detects camouflage but suffers six false positive (Table 5.4), as a result of the target cutting across the corner of the interfering object, causing the particles to be transferred into the vicinity of the camouflaging object during normal tracking.
- Approach (2) and (3) both detects camouflage with no errors (Table 5.5 and Table 5.6 respectively).

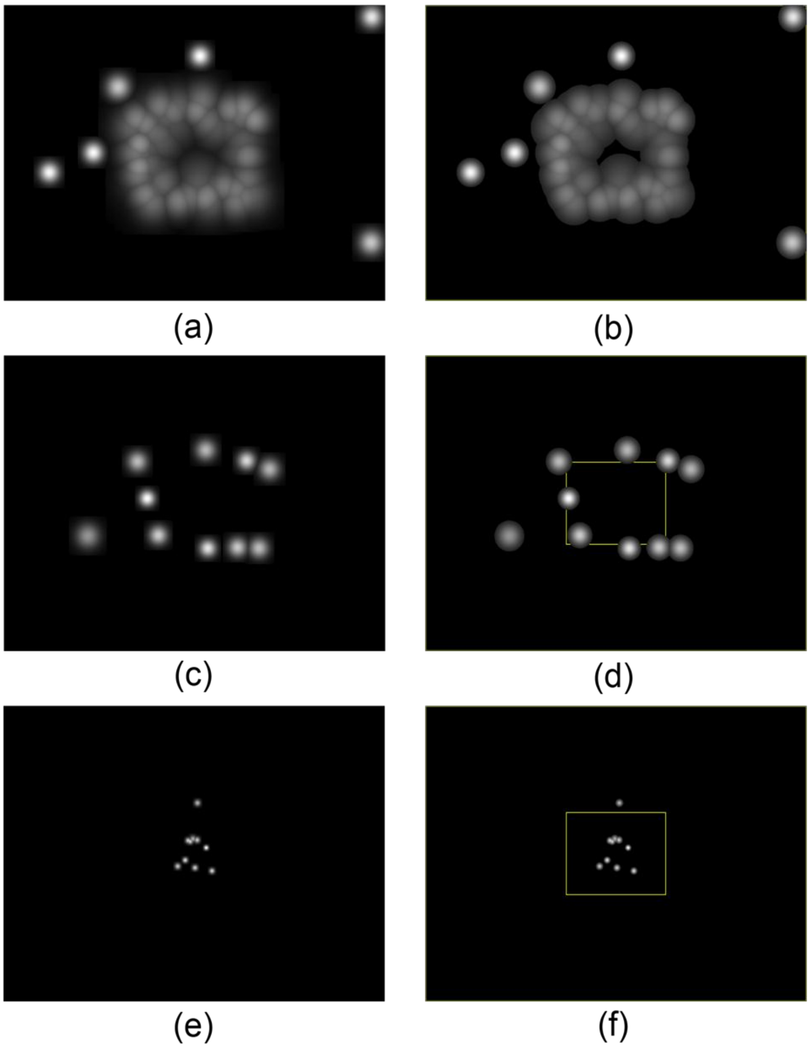


Figure 5.5 Camouflage maps for artificial camouflage scene mapping videos via: (a) approach (1), (c) approach (2), (e) approach (3); (b), (d), and (f) shows camouflage maps overlaid on edge-detected images of the viewed scene.

Camouflage maps generated for artificial camouflage scene mapping shows that:

- All three approaches correspond to areas in the environment where camouflage occurs.
- Maps produced for approach (1) (Figure 5.5 (a), (b)) and approach (3) (Figure 5.5 (e), (f)) shows entries are gathered inside the camouflaging object boundary. While, approach (2) produced maps ((Figure 5.5 (c), (d)) showing entries are gathered outside or on the boundaries of the camouflaging object. Approach (2) fires outside or on the boundaries of the camouflaging object is due to regions in the PBI, expanded as a result of camouflage, thus, causing the detector to fire.

Table 5.7 Confusion matrix for real occlusion scene mapping videos using approach (1): particle clustering.

		Predicted	
		Normal tracking	*Occlusion*
Actual	*Occlusion*	2	**142**

Table 5.8 Confusion matrix for real occlusion scene mapping videos using approach (2): particle spread measured relative to the PBIs medial axis.

		Predicted	
		Normal tracking	*Occlusion*
Actual	*Occlusion*	2	**18**

Table 5.9 Confusion matrix for real occlusion scene mapping videos using approach (3): alignment of the PBI medial axis and tracker's path estimate.

		Predicted	
		Normal tracking	*Occlusion*
Actual	*Occlusion*	5	**15**

The confusion matrices for real occlusion scene mapping leads to the following summary:

- All three approaches are capable of detecting occlusion events, though all three approaches record some misclassifications.
- Approach (1) and (2) suffers two false negative (Table 5.7 and Table 5.8 respectively); while approach (3) performed less well and suffers five false negative (Table 5.9). These misclassifications are caused by the detector firing incorrectly due to the targets being tracked here are of variable size and most are wider than the occluding object.

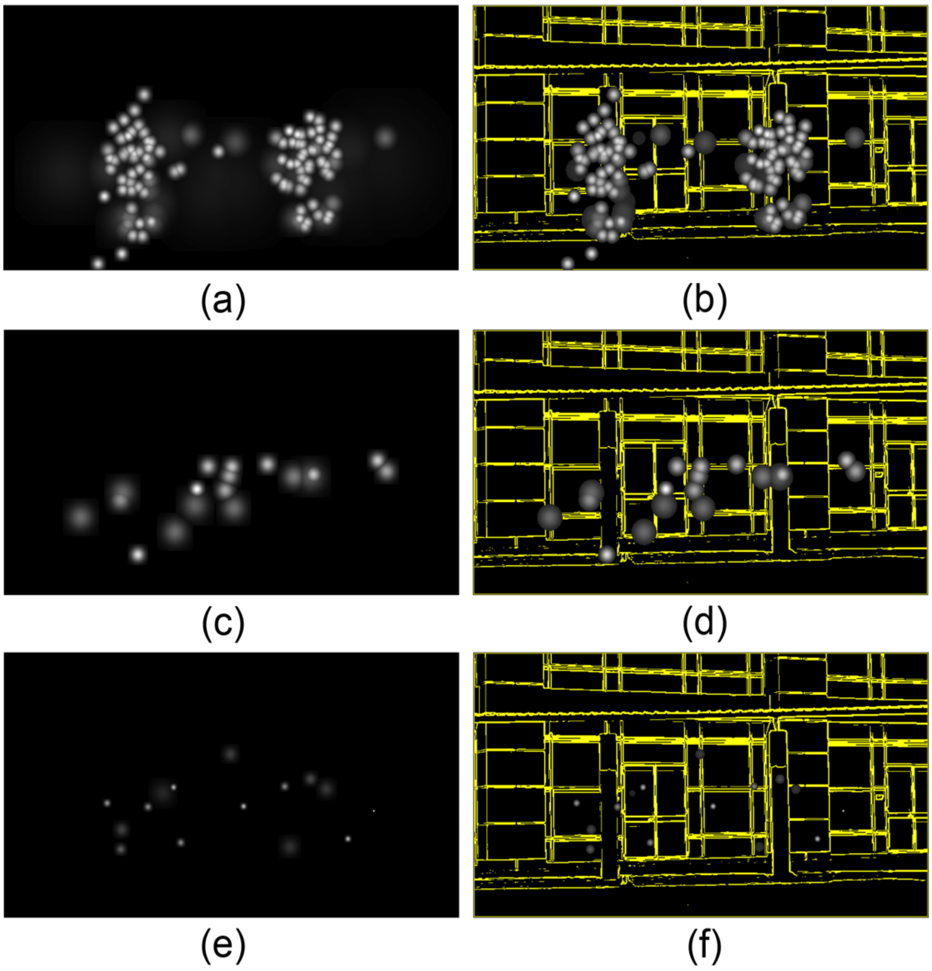


Figure 5.6 Occlusion map for real occlusion scene mapping videos via: (a) approach (1), (c) approach (2), (e) approach (3); (b), (d), and (f) show occlusion maps overlaid on edge-detected images of the viewed scene.

Analysing the occlusion maps for real occlusion scene mapping provides evidence that:

- Maps produced showed all three approaches correspond to the areas in the environment where the two narrow vertical pillars are located, which causes occlusion to occur.
- Approach (1) produced maps (Figure 5.6 (a), (b)) whose entries marked the narrow vertical pillars that cause occlusion. Approach (2) (Figure 5.6 (c), (d)) and approach (3) (Figure 5.6 (e), (f)) also produced maps showing entries gathered around the pillars. It should be noted that for approach (2) and (3) respectively, most entries are gathered in between the vicinity of the two pillars because these two approaches rely on spatial information computed from the medial axis of the PBI, whereas approach (1) rely on particle clusters formed independently, at each time step.

- Note that although the pillar boundaries are less clear than in the maps for artificial scene mapping videos, this is because the targets tracked here are of variable size and more, importantly, most are wider than the occluding pillars.

In summary:

1. The confusion matrices show that all three approaches successfully distinguish targets moving normally or experiencing occlusion or camouflage. Table 5.10 presents the rate of accuracy (%) of each approach and is computed from the confusion matrices presented earlier in this chapter. Rate of accuracy is defined in (5.1) as the number of test videos that were correctly interpreted divided by the total number of videos in the test set. Here, correctly interpreted means that a disruptive event was correctly identified (true positive) or no events were reported during normal tracking (true negative).

$$Rate\_of\_accuracy\_(\%) = \left(\frac{Number\_of\_correct\_interpretations}{Number\_of\_samples}\right) \times 100 \qquad (5.1)$$

Table 5.10 Event detection accuracy (% correct interpretations) of each approach.

		Scene mapping type		
		Artificial occlusion	*Artificial camouflage*	*Real occlusion*
Methods	*Approach (1)*	97%	94%	98.6%
	Approach (2)	93%	100%	90%
	Approach (3)	80%	100%	75%

2. Examination of Table 5.10 and the data presented above suggests that:

   - In artificial occlusion scene mapping, approaches (1) and (2) recorded an accuracy rate of 97% and 93% respectively. Approach (3) meanwhile only recorded an 80% accuracy rate. Approaches (1) and (2) suffer a 3% and 7% loss, respectively, in accuracy because the detector fired incorrectly during normal tracking, when the target cuts across the corner of the occluding object. Again, using a larger moving window size can improve the performance of approaches (1) and (2). Approach (3) suffers a 20% loss in accuracy due to noise on the path which causes the detector to fire/not fire correctly. Therefore, we hypothesise that removing these effects can significantly improve the performance of approach (3).

- In artificial camouflage scene mapping, approach (2) and (3) performed far better than approach (1). Both approaches have an accuracy rate of 100% at detecting camouflage as compared to approach (1) which has an accuracy rate of 94%. Approach (1) suffers a 6% loss in accuracy due to particles being transferred into the vicinity of the camouflaging object when the target cuts across the corner of the interfering object, causing the detector to fire incorrectly. We hypothesise that the use of a larger moving window size can make the detector less sensitive to these instances and improve the performance of approach (1).
- In real occlusion scene mapping, approach (1) recorded the highest accuracy rate at 98.6%, followed by approach (2) with 90% and approach (3) with 75%. The loss in accuracy for all three approaches can be due from the detector firing incorrectly as a result of tracking targets of variable size and most are wider than the occluding object. Therefore, using better targets can help improve the performance of all three approaches.
- In scene mapping, approaches (2) and (3) perform better than approach (1) in artificial camouflage but do not perform as well as approach (1) in artificial and real occlusion. It should be stressed here that comparing approach (2) and (3) with approach (1) is made more difficult due to the different number of samples used. We hypothesise that the use of similar number of samples as in approach (1) will improve the detection accuracy of both approach (2) and (3), respectively. The current data, however, shows that approach (2) is better than approach (3) in all scene mapping scenarios.

3. All three approaches produced maps that identified locations in the viewed environment where disruptive events occurred as well as marked the boundaries of these camouflaging/occluding objects.

5.5. Chapter summary

Literature has shown that scene map plays an integral part in a number of different research fields.

Three different approaches were applied to sets of scene mapping videos and the results produced from each approach exploited using process-behaviour chart to detect occurrences of occlusion and camouflage. These were subsequently used to create image-based scene maps marking locations at which each event occurred. Analysis of results provided evidence that all three approaches are equally and very successful at differentiating between a target moving normally or experiencing occlusion or camouflage. Analysis also demonstrated that the scene maps created via each approach correspond to areas of the background environment in which occlusion and/or camouflage are likely to occur.

This opens ups the possibility of the tracker being able to detect and respond appropriately to inevitable, highly disruptive events.

Chapter 6 Contributions and Future Work

6.1. Contributions

The primary contribution of the work reported here is the demonstration that the distribution and behaviour of the particle sets generated by a particle filter-based tracker carry information about the current state of the tracking process, and can be used to detect occlusion and camouflage events. Detection is achieved via the novel application of the Shewhart process-behaviour chart, a method developed within control engineering to monitor the internal state of a process.

The control chart is applied to three distinct measures of particle behaviour:

1. The standard deviation of the particle clusters formed during particle filter-based tracking.
2. Measurements of particle spread made after aggregating the particle sets generated over a temporal window.
3. Estimates of the errors in the alignment of the medial axis of the aggregated particle sets and the estimated target path as provided by the tracker.

A desire to aggregate particle sets over time lead to the second contribution of the research: the formulation of a novel view-based representation of visual motion, the Particle History Image, and a related data structure capturing the boundary of the aggregated particles, the Particle Boundary Image. Though the focus here is on the detection of occlusion and camouflage events, it is anticipated that these representations will find further application elsewhere.

The third contribution of the work is an experimental evaluation of the new techniques. The proposed methods have been applied to real and artificial image sequences, their ability to detect disruptive events assessed and the output of the process-behaviour charts used to create Gaussian Mixture Models of the occlusion and camouflage structure of the underlying scene. Examination of the resulting data leads to the observations that:

- All three representations allowed successful event detection.
- Performance was increased when the process-behaviour control charts were constructed using longer training sequences.

- Though the particle clustering and PBI medial axis methods are capable of distinguishing occlusion and camouflage, the path alignment method is not. It does however, provide an indication that tracking is being lost.
- Evaluation of the three detection methods showed that the medial axis detection method is the best detection method. The method is less sensitive to false data and firing prematurely. We suggest that this is due to the PHI providing a better representation of particle set behaviour by aggregating particles over a longer time period. The use of a training set to parameterise the process-behaviour control chart means that comparisons are being made between measurements that are both made over extended time periods, and so more reliable.
- The presence of noise affected the tracker's path estimate more severely when measuring the width of the occupied region of a PHI as compared to measuring the instantaneous particle spread, independently, at each time step.
- In scene mapping, evaluation of the three different approaches showed that all three approaches created maps that outlined the true silhouette of the camouflaging/occluding object as well as the boundaries of these interfering objects in a viewed environment. It is anticipated that these maps will find further application in trackers created to detect and respond appropriately to these disruptive events.
- Note that the size of the target affects the event detection accuracy of each approach. We hypothesise that this is due to the target size affecting the behaviour of the particles which tracks them. Further investigation into this behaviour is beyond the scope of this thesis, though it does open up new questions about the relationship between segmentation and particle-based algorithms, and possible future work.

The contributions mentioned above successfully fulfil the objectives specified in section 1.2. A discussion on self-occlusion and 3D motion is given in section 6.2 and possible extensions of the work reported here are discussed in section 6.3.

6.2. Self-occlusion and 3D

Results in Chapter 3 and Chapter 4 showed the particle clustering method to be the best detection method at handling self-occlusion. The PBI medial axis method could not detect self-occlusion successfully as no clear motion was recorded due to the particle set(s) generated at newer time step overlapped over particle set(s) from prior time steps, thus, generating a blob like PHI and PBI. This resulted in no medial axis path being recorded. With no medial axis path, the alignment distance method could not be used to detect for self-occlusion.

In Chapter 3, the particle cluster deviation (spread) analysis showed that the deviation increases when the particle spread out across the camouflaging object as camouflage occurs and decreases when the particle spread becomes closely clustered as a result of occlusion. These events can increase/decrease the estimated (and so the predicted) speed. Currently, deviation is used as the main indicator, while speed acts as a secondary indicator in the detection of occlusion and camouflage. This makes the particle clustering method more general as other factors (e.g. size of the target) is assumed to be fixed throughout. We hypothesise that by extending the evaluation to 3D using the target size, speed and deviation can improve the accuracy of the detection method at detecting occlusion and camouflage.

6.3. Future Work

This section explores possible improvements and further investigation to the research work documented in this thesis.

6.3.1. Detecting occlusion and/or camouflage in 3D motion

The work presented here focuses primarily on occlusion and camouflage occurring when tracking target(s) moving in two-dimension. Results provided evidence that the detection of occlusion and/or camouflage can be done by exploiting the results gathered from either one of three approaches documented in this thesis using process-behaviour chart.

Further investigation to find a solution for detecting these disruptive events in three-dimension is beyond the scope of this thesis; nevertheless, it does lead to future work. Possible solution to this problem could lie in:

1. Performing additional experiments to understand how the results gathered from applying the particle clustering method can be further improved prior to exploiting the results using process-behaviour chart.
2. Evaluating more information about the target (e.g. monitoring the change in the target’s shape) during 3D motion.

6.3.2. Automating tools

In Chapter 4, improving the results of the non-thinned PBIs and the medial axis are done via manual editing as the creation of automated tools is beyond the scope of this thesis. Thus, future work of creating algorithms to perform these improvements automatically is feasible.

In the improvement of the non-thinned PBIs, an algorithm can be created which analyses the graphical representation of the non-thinned PBI and identify existing gaps within the non-thinned PBI boundary which is later filled and closed.

As for the improvement of the medial axis path, a separate algorithm can be created which analyses along the PBI medial axis path and identify point(s) that does not fit within the optimum path of the medial axis, resulting in that point(s) being removed.

6.3.3. Improvements to the texture edge detection algorithm

At present, there is a short delay in the detection of disruptive events as a result of constructing PHI and PBI using the texture edge detection algorithm presented in Chapter 4. Future work is required to investigate how the performance of the texture edge detection algorithm can be improved so as to allow the construction of PHI and PBI to be done without any delay.

One possible way to improve the performance of the texture edge detection algorithm is during the Compass operator process. (Maxwell et al., 2003) suggested taking a sampling approach to estimating the texture gradient in the image as an alternative to computing Compass operator at each pixel of the 14 Laws’ texture feature images.

6.3.4. Using alternative scene mapping videos

For scene mapping analysis detailed in Chapter 5, only three sets of test videos were used. Analysis showed that even by using three sets of test videos, the three approaches successfully detected and mapped out occlusion and camouflage.

Further investigation can involve applying the three approaches to different and more complex sets of scene mapping videos and analysing the results it produce when detecting and mapping out occlusion and/or camouflage.

6.3.5. Improving tracking

The work detailed here presented three approaches to detect and subsequently, map out the occurrences of occlusion and camouflage during tracking. Further investigation into how trackers can utilise the results produced by these three approaches to respond appropriately after experiencing occlusion and camouflage is beyond the scope of this thesis.

Therefore, future work is required to exploit the knowledge gained from the investigation presented here to create an improved particle filter-based tracker that is capable of detecting and responding appropriately to these disruptive events.

Appendix A **Gaussian Mixture Model**

A.1. 2 Dimension EM Algorithm

```
GAUSSIAN MIXTURE MODEL ALGORITHM

1) Assign $N$ number of particles, $\{X_t^{(n)}\}_{n=1}^{N}$ to $g$ number of clusters. Clusters' mean, deviation and mixture weight are randomly computed during initialization.
2) Compute mixture model by considering all particles in all clusters':
       $$P(x) = \sum_{i}^{g} \pi_i P(x, y \mid \theta_i)$$
3) Perform an iterative Expectation Maximization (EM) algorithm:
       $$P(\theta_i \mid x) = \frac{P(x \mid \theta_i) \cdot P(\theta_i)}{P(x)}$$
4) Re-estimate cluster parameters at iteration step, $s+1$by taking into consideration the probability of each particle belonging to each cluster.
5) Check for convergence between clusters at iteration step,$s$ and$s+1$using the Euclidean distance formula:
       $$\sqrt{\left(\mu_x^{s+1} - \mu_x^{s}\right)^2 + \left(\mu_y^{s+1} - \mu_y^{s}\right)^2} \leq 1.0$$
   a. If no convergence is attained, then replace the clusters at iteration step, $s$ with the clusters from iteration step, $s+1$ and repeat from step 2;
   b. If convergence is attained, classify particles based on the cluster with the highest Gaussian probability value.
       i. Recalculate the cluster parameters and weights after the particles have been split into their respective clusters.
       ii. Feed the clustered particles to the particle filtered-tracker.
```

Algorithm A.1 Gaussian Mixture Model algorithm.

Particles are defined as $\left\{\left(X_t^n\right), n = 1, \ldots, N\right\}$, where X_t^n is the target state vector representations in coordinate location-(x, y), *n* is the *n*th particle set, *t* is the time step and *N* is the sample size of the population. At each time step, the sample size of the particle population is equally assigned to each cluster. The cluster's mean for coordinate location-(x, y) are weighted using (A.1) and (A.2), while deviation values are weighted using (A.3). The mixture weight is computed using

(A.4). The cluster's mean, deviation and mixture weight for the initial iteration are randomly selected. For successive iterations, a weighted mean and a weighted deviation are computed.

$$\mu_x = \frac{\sum_{i=1}^{N} P(\theta_i \mid x, y) x_i}{\sum_{i=1}^{N} P(\theta_i \mid x, y)} \tag{A.1}$$

$$\mu_y = \frac{\sum_{i=1}^{N} P(\theta_i \mid x, y) y_i}{\sum_{i=1}^{N} P(\theta_i \mid x, y)} \tag{A.2}$$

$$\sum{}_{x,y} = \sqrt{\frac{\sum_{i=1}^{N} P(\theta_i \mid x, y)\left(\sqrt{(x_i - \mu_x)^2 + (y_i - \mu_y)^2}\right)^2}{\sum_{i=1}^{N} P(\theta_i \mid x, y)}} \tag{A.3}$$

$$\pi_c = \frac{\sum_{m=1}^{M} w_m^{(c)}}{\sum_{i=1}^{N} w_i} \tag{A.4}$$

N is the particle sample size population, w_i is the clusters' weight, $w_m^{(c)}$ is the cluster's weight, *M* is the particle sample size population within a single cluster and $P(\theta_i \mid x, y)$ is the probability of the particle being in a cluster. The mixture model (A.5) is then computed by taking all particles in every cluster into concern.

$$P(x, y) = \sum_{i}^{g} \pi_i P(x, y \mid \theta_i) \tag{A.5}$$

π_i is the mixing weights, *g* is the total number of clusters and $P(x, y \mid \theta_i)$ is a probability density function of particles' from the cluster's mean. Probability density function is computed using (A.6):

$$\frac{1}{\sqrt{(2\pi)}\sqrt{(\Sigma_{x,y})}} \exp\left(- \frac{\left(\sqrt{(x - \mu_x)^2 + (y - \mu_y)^2}\right)^2}{2\Sigma_{x,y}} \right) \tag{A.6}$$

The **Expectation** step is performed by using the Bayes' rule (Bayes et al., 1763):

$$P(X \mid Y) = \frac{P(Y \mid X) P(X)}{P(Y)} \tag{A.7}$$

$P(X\mid Y)$ is the posterior distribution, $P(Y\mid X)$ is the conditional probability likelihood of *Y* given *X*, $P(X)$ is the prior distribution and $P(Y)$ is the normalization factor which guarantees that $P(X\mid Y)$ amounts to unity. Bayes' rule allows the posterior distribution to be inferred from observation data. Therefore, the probability of each particle arising from each cluster is computed:

$$P(\theta_i \mid x,y)=\frac{P(x,y\mid\theta_i)P(\theta_i)}{P(x,y)} \tag{A.8}$$

$P(x,y\mid\theta_i)$ is the probability of a cluster generating each particle, $P(\theta_i)$ is the cluster weight and $P(x,y)$ is the mixture model.

In the **Maximization** step, improved parameters for each cluster at iteration step $s+1$ are computed by taking into consideration the probability of each particle being in each cluster. The convergence of the EM algorithm is based on the Euclidean distance (A.9) between clusters at iteration step $s+1$ and s.

$$\sqrt{\left(\mu_x^{s+1}-\mu_x^{s}\right)^2+\left(\mu_y^{s+1}-\mu_y^{s}\right)^2} \tag{A.9}$$

When the EM algorithm has successfully converged, particles are assigned to mixture components based on the cluster with the highest probability value and the clusters' parameters are recomputed.

A.2. 3 Dimension EM Algorithm

A revised mathematical representation for a three dimension EM algorithm is given in (A.10) to (A15). The weighted means are computed using (A.10), (A.11) and (A.12), while (A.13) computes the weighted deviation. The revised mixture model is computed using (A.14) and the Euclidean distance to determine EM convergence is computed using (A.15).

$$\mu_x=\frac{\sum_{i=1}^{N}P(\theta_i\mid x,y,\Pi)x_i}{\sum_{i=1}^{N}P(\theta_i\mid x,y,\Pi)} \tag{A.10}$$

$$\mu_y=\frac{\sum_{i=1}^{N}P(\theta_i\mid x,y,\Pi)y_i}{\sum_{i=1}^{N}P(\theta_i\mid x,y,\Pi)} \tag{A.11}$$

$$\mu_{\Pi} = \frac{\sum_{i=1}^{N} P(\theta_i \mid x, y, \Pi)\Pi_i}{\sum_{i=1}^{N} P(\theta_i \mid x, y, \Pi)} \tag{A.12}$$

$$\sum{}_{x,y,\Pi} = \sqrt{\frac{\sum_{i=1}^{N} P(\theta_i \mid x, y, \Pi)\left(\sqrt{(x_i - \mu_x)^2 + (y_i - \mu_y)^2 + (\Pi_i - \mu_{\Pi})^2}\right)^2}{\sum_{i=1}^{N} P(\theta_i \mid x, y, \Pi)}} \tag{A.13}$$

$$P(x, y, \Pi) = \sum_{i}^{g} \pi_i P(x, y, \Pi \mid \theta_i) \tag{A.14}$$

$$\sqrt{\left(\mu_x^{s+1} - \mu_x^{s}\right)^2 + \left(\mu_y^{s+1} - \mu_y^{s}\right)^2 + \left(\mu_{\Pi}^{s+1} - \mu_{\Pi}^{s}\right)^2} \tag{A.15}$$

Appendix B Number of Clusters Analysis

This appendix presents the results gathered from applying different number of clusters on the remaining test videos presented in Chapter 2.

B.1. Occlusion

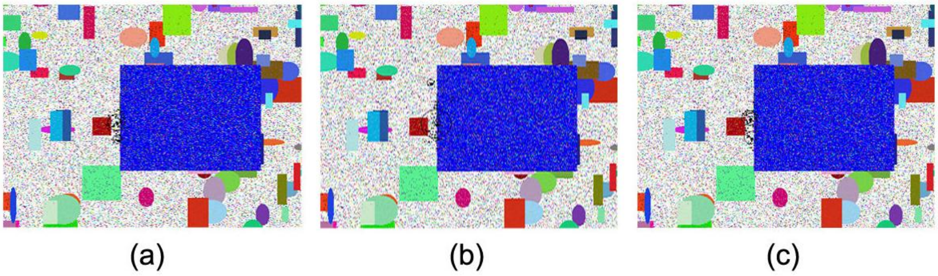


Figure B.1 Tracking a yellow circle experiencing occlusion amid clutter when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

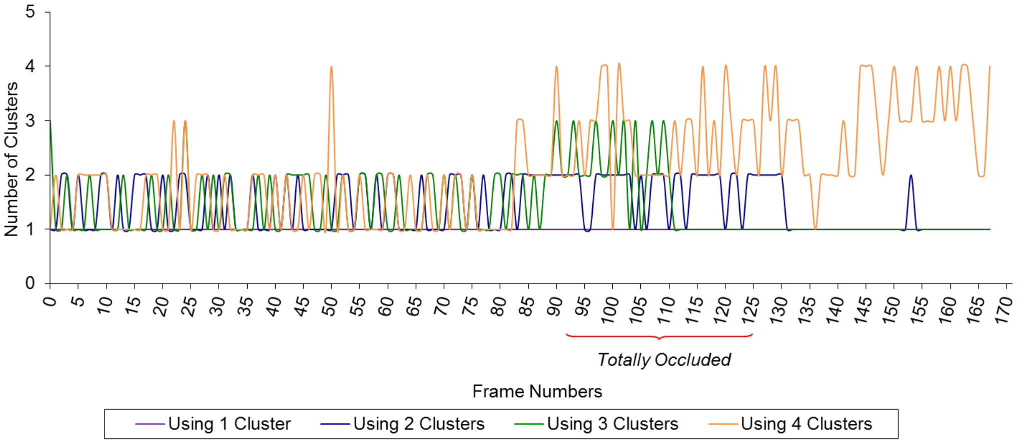


Figure B.2 Graph showing the use of different numbers of clusters for a yellow circle experiencing occlusion amid clutter.

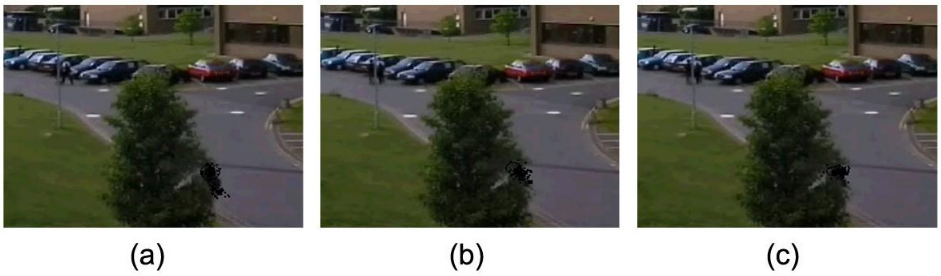


Figure B.3 A tree occluding a woman when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

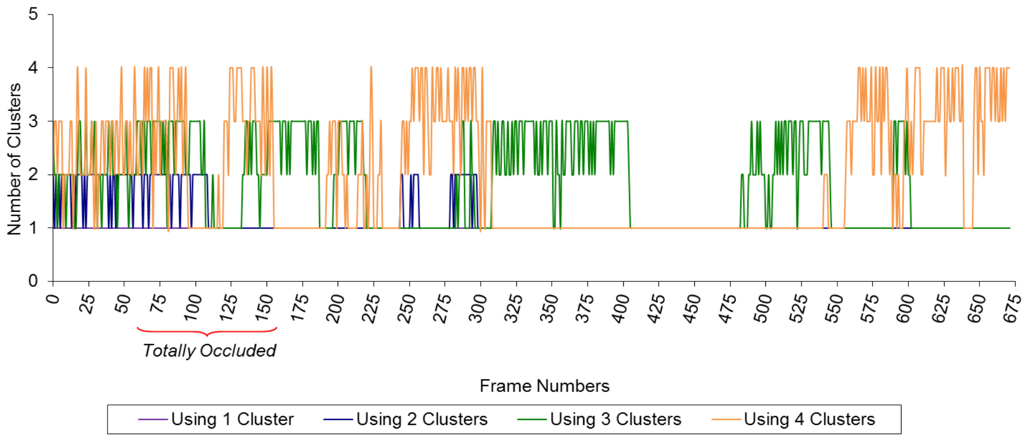


Figure B.4 Graph showing the use of different numbers of clusters for a tree occluding a woman.

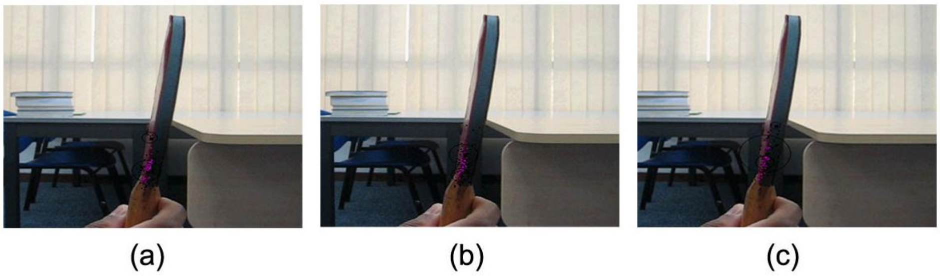


Figure B.5 A table tennis bat experiencing self-occlusion when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

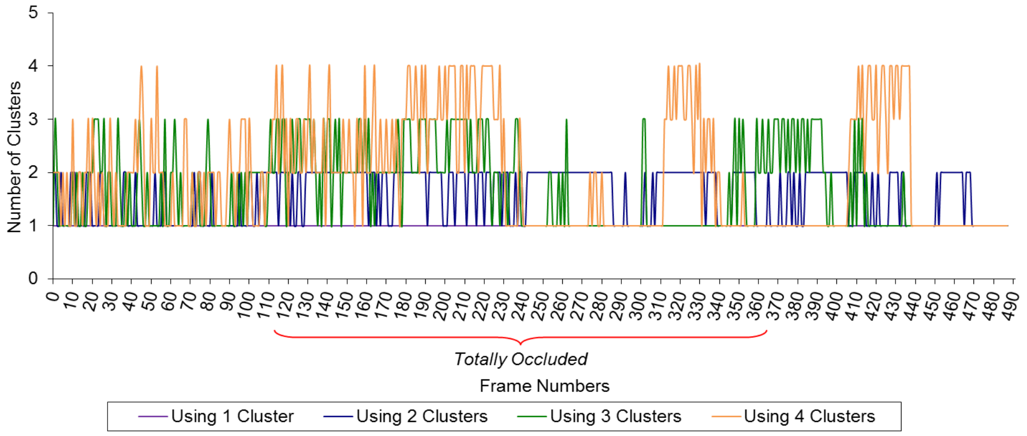


Figure B.6 Graph showing the use of different numbers of clusters for a table tennis bat experiencing self-occlusion.

B.2. Camouflage

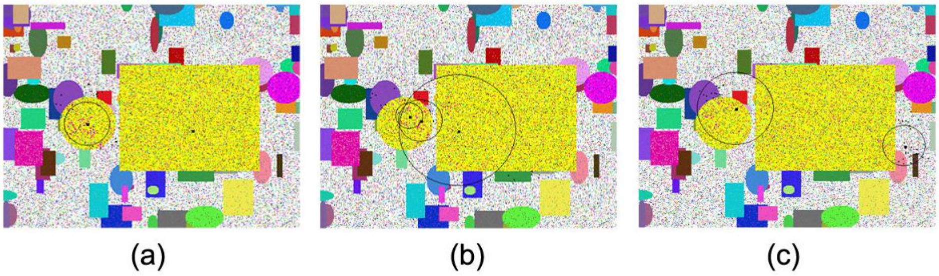


Figure B.7 Tracking a yellow circle experiencing camouflage amid clutter when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

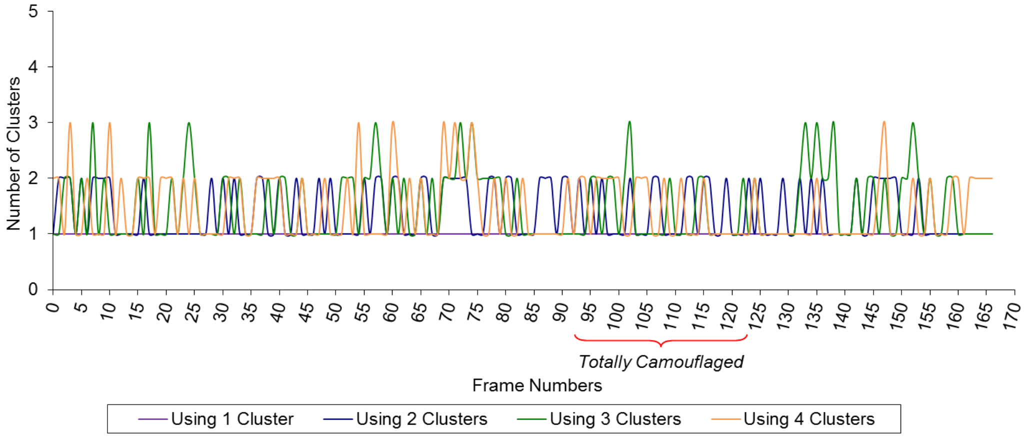


Figure B.8 Graph showing the use of different numbers of clusters for a yellow circle experiencing camouflage amid clutter.

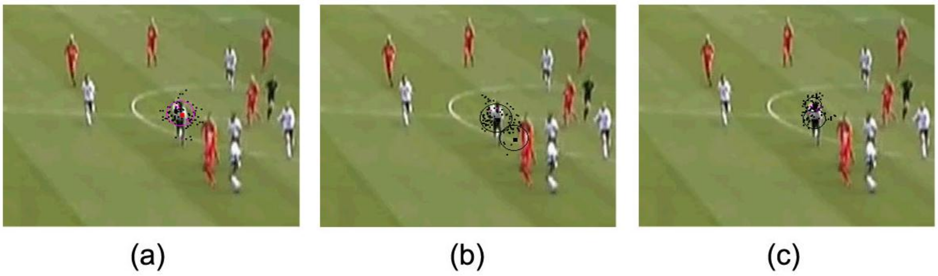


Figure B.9 Tracking a football being camouflaged by a player when using: (a) 2 clusters, (b) 3 clusters or (c) 4 clusters.

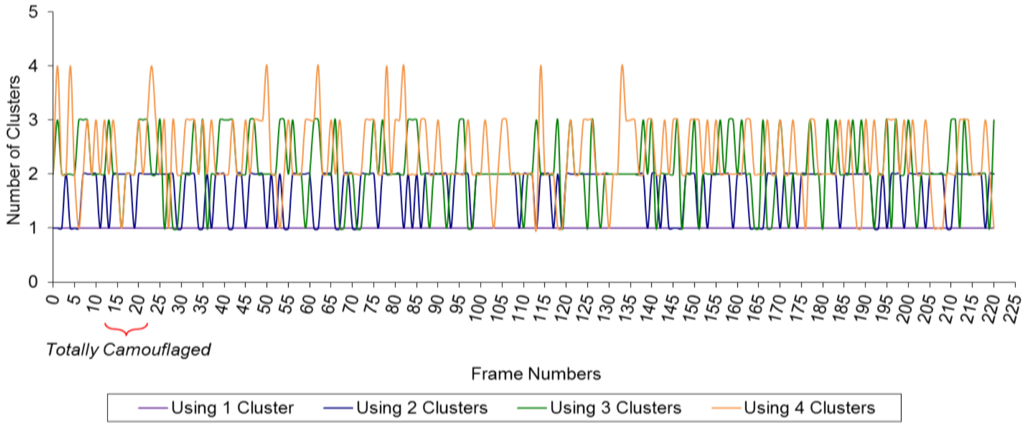
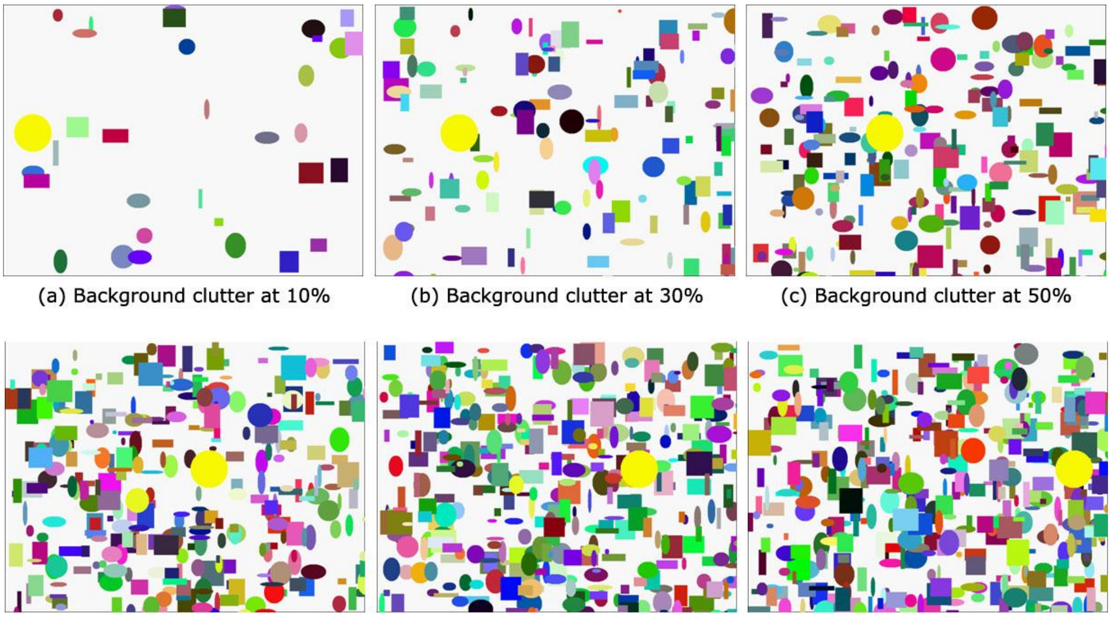


Figure B.10 Graph showing the use of different numbers of clusters for a football being camouflaged by a player.

Appendix C Evaluation

C.1. Normal tracking



(a) Background clutter at 10%

(b) Background clutter at 30%

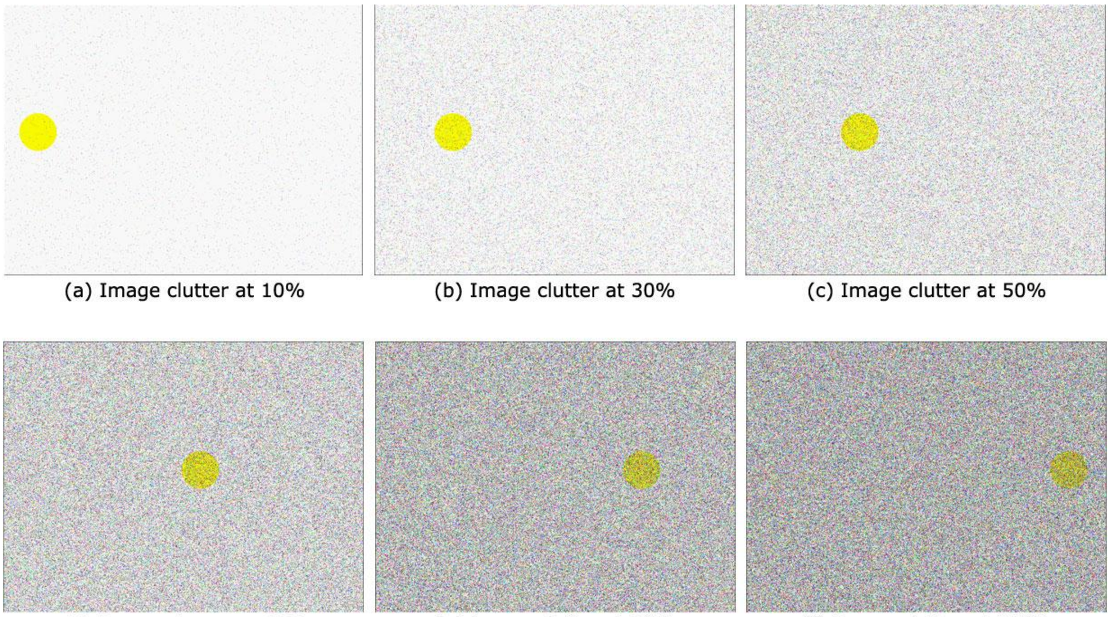
(c) Background clutter at 50%

(d) Background clutter at 70%

(e) Background clutter at 90%

(f) Background clutter at 100%

Figure C.1 Different percentage levels of background clutter with the minimum value being 10% and the maximum value being 100%.



(a) Image clutter at 10%

(b) Image clutter at 30%

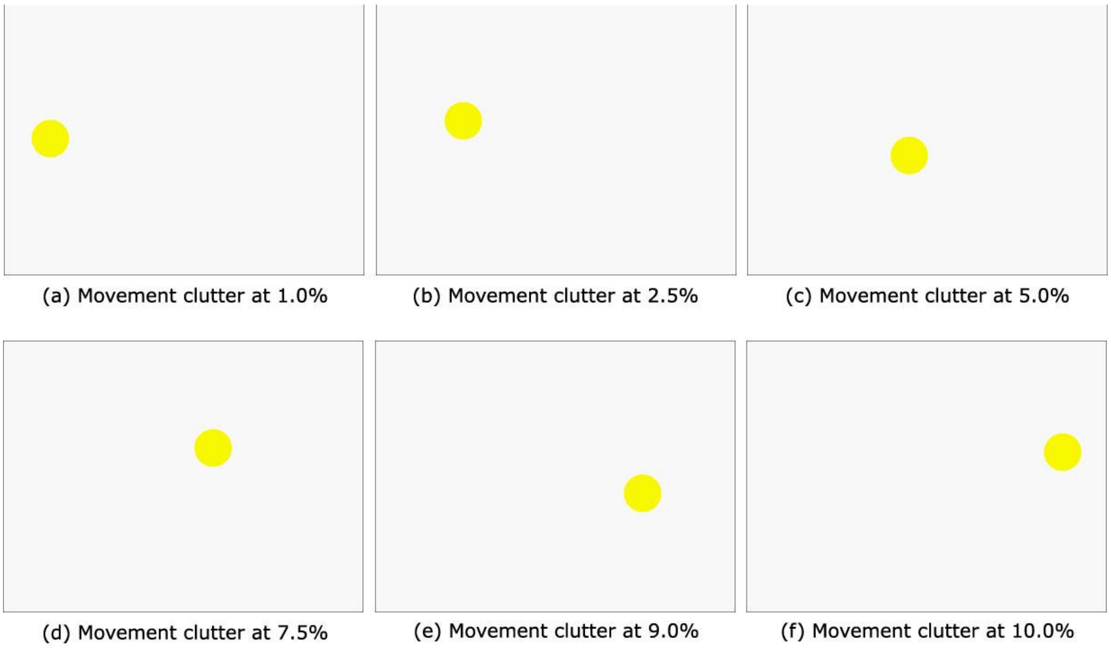
(c) Image clutter at 50%

(d) Image clutter at 70%

(e) Image clutter at 90%

(f) Image clutter at 100%

Figure C.2 Different percentage levels of image clutter with the minimum value being 10% and the maximum value being 100%.



(a) Movement clutter at 1.0%

(b) Movement clutter at 2.5%

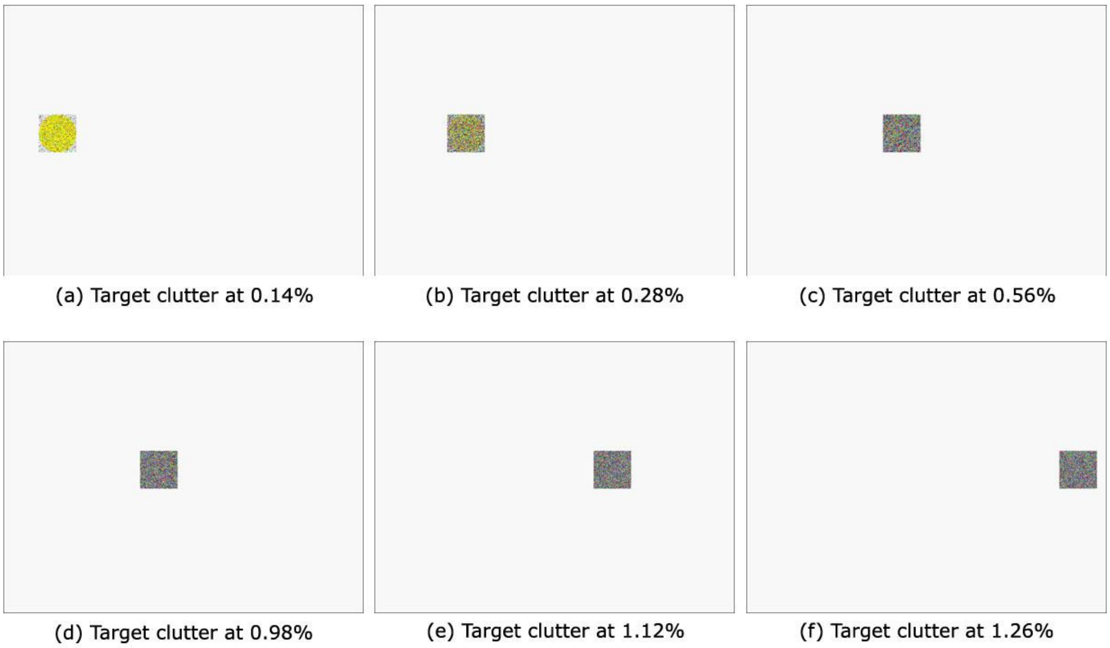
(c) Movement clutter at 5.0%

(d) Movement clutter at 7.5%

(e) Movement clutter at 9.0%

(f) Movement clutter at 10.0%

Figure C.3 Different percentage levels of motion clutter with the minimum value being 1.0% and the maximum value being 10.0%.



(a) Target clutter at 0.14%

(b) Target clutter at 0.28%

(c) Target clutter at 0.56%

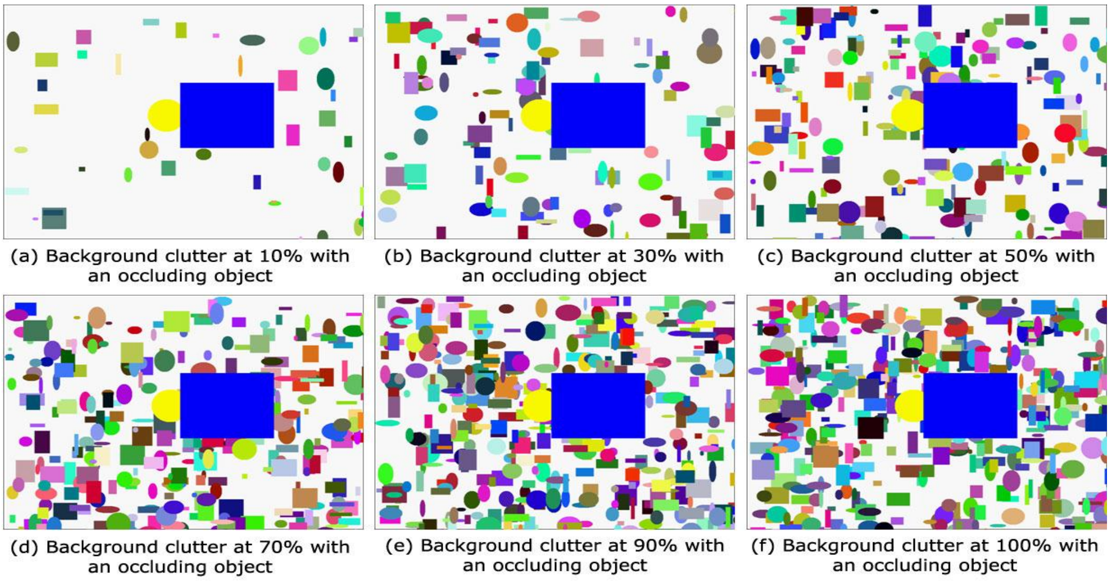
(d) Target clutter at 0.98%

(e) Target clutter at 1.12%

(f) Target clutter at 1.26%

Figure C.4 Different percentage levels of target clutter with the minimum value being 0.14% and the maximum value being 1.26%.

C.2. Occlusion


(a) Background clutter at 10% with an occluding object

(b) Background clutter at 30% with an occluding object

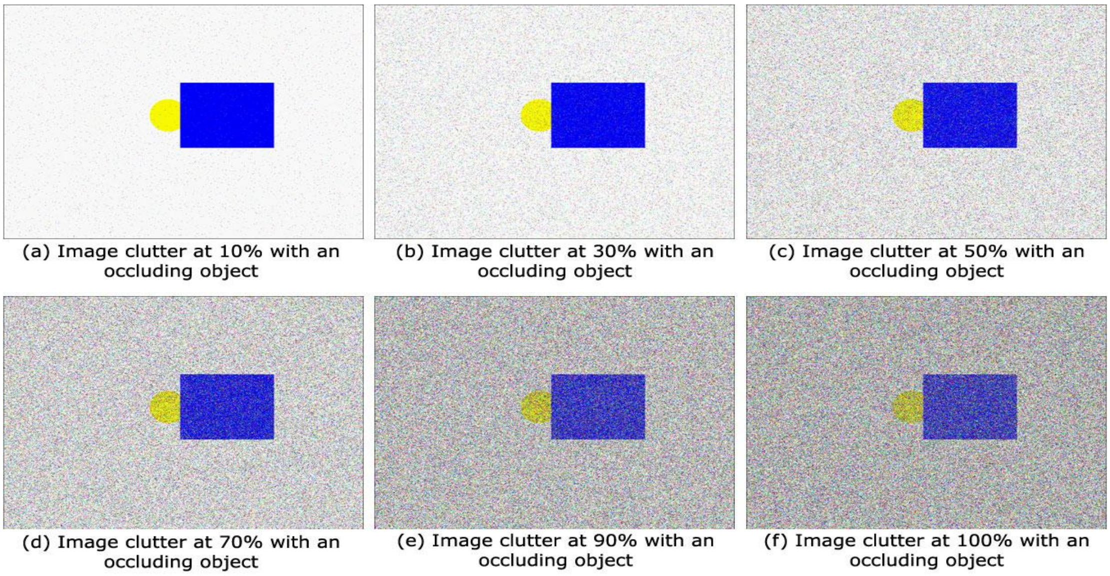
(c) Background clutter at 50% with an occluding object

(d) Background clutter at 70% with an occluding object

(e) Background clutter at 90% with an occluding object

(f) Background clutter at 100% with an occluding object

Figure C.5 Different percentage levels of background clutter during the occurrence of occlusion with the minimum value being 10% and maximum value being 100%.


(a) Image clutter at 10% with an occluding object

(b) Image clutter at 30% with an occluding object

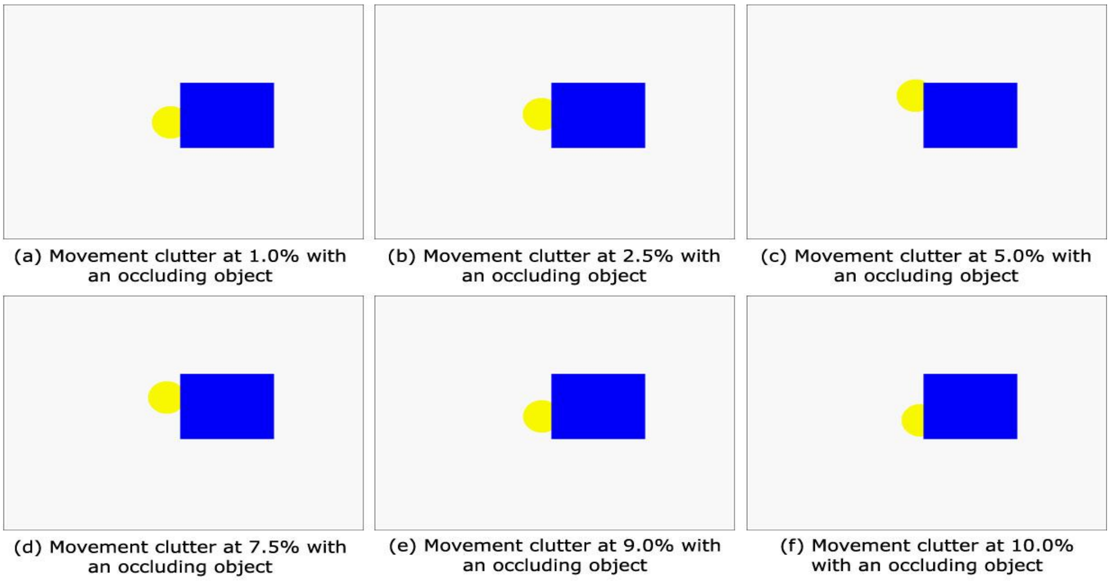
(c) Image clutter at 50% with an occluding object

(d) Image clutter at 70% with an occluding object

(e) Image clutter at 90% with an occluding object

(f) Image clutter at 100% with an occluding object

Figure C.6 Different percentage levels of image clutter during the occurrence of occlusion with the minimum value being 10% and maximum value being 100%.



(a) Movement clutter at 1.0% with an occluding object

(b) Movement clutter at 2.5% with an occluding object

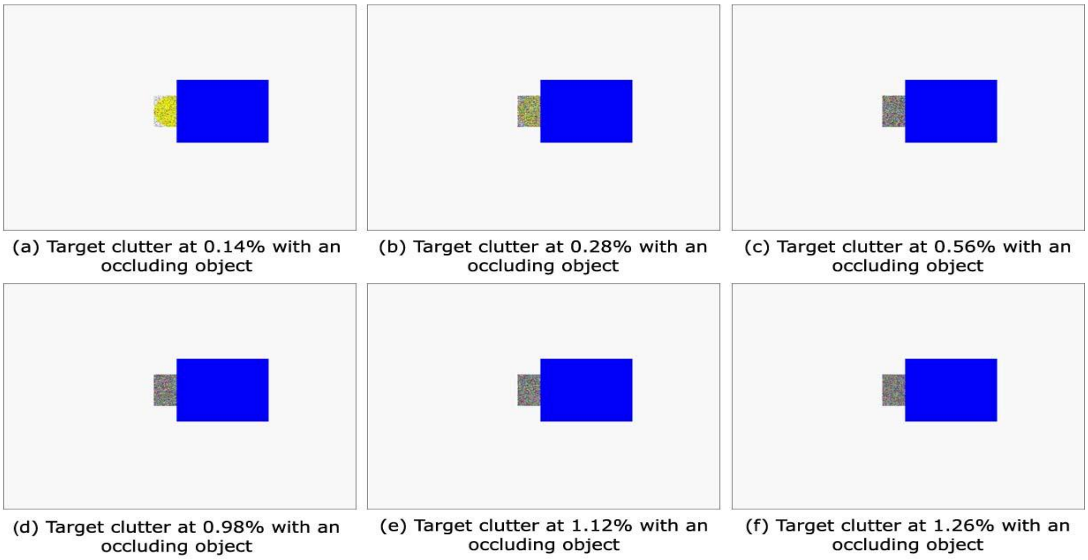
(c) Movement clutter at 5.0% with an occluding object

(d) Movement clutter at 7.5% with an occluding object

(e) Movement clutter at 9.0% with an occluding object

(f) Movement clutter at 10.0% with an occluding object

Figure C.7 Different percentage levels of motion clutter during the occurrence of occlusion with the minimum value being 1.0% and maximum value being 10.0%.



(a) Target clutter at 0.14% with an occluding object

(b) Target clutter at 0.28% with an occluding object

(c) Target clutter at 0.56% with an occluding object

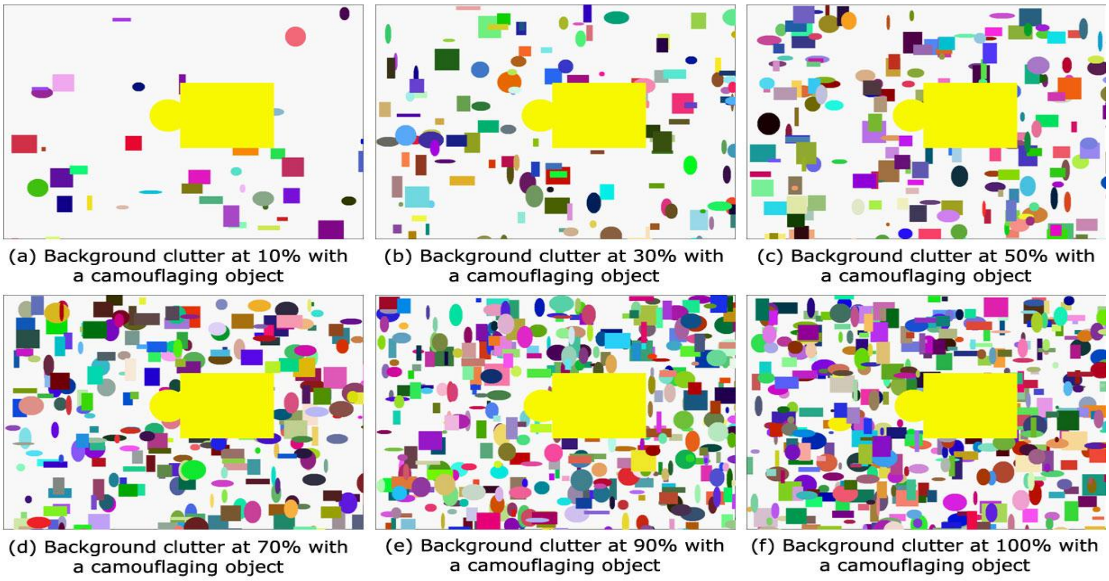
(d) Target clutter at 0.98% with an occluding object

(e) Target clutter at 1.12% with an occluding object

(f) Target clutter at 1.26% with an occluding object

Figure C.8 Different percentage levels of target clutter during the occurrence of occlusion with the minimum value being 0.14% and maximum value being 1.26%.

C.3. Camouflage



(a) Background clutter at 10% with a camouflaging object

(b) Background clutter at 30% with a camouflaging object

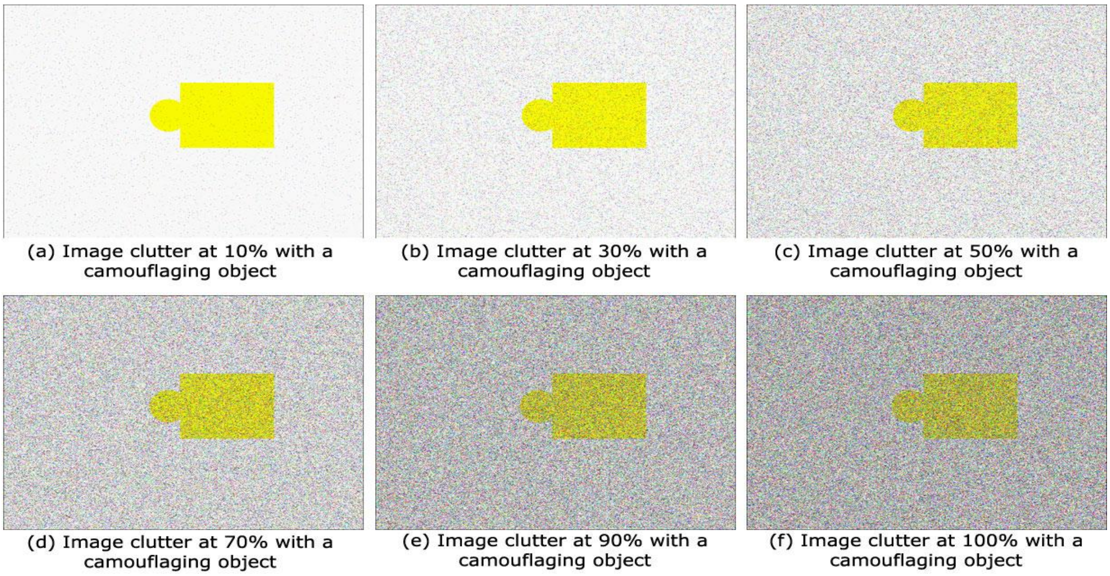
(c) Background clutter at 50% with a camouflaging object

(d) Background clutter at 70% with a camouflaging object

(e) Background clutter at 90% with a camouflaging object

(f) Background clutter at 100% with a camouflaging object

Figure C.9 Different percentage levels of background clutter during the occurrence of camouflage with the minimum value being 10% and maximum value being 100%.



(a) Image clutter at 10% with a camouflaging object

(b) Image clutter at 30% with a camouflaging object

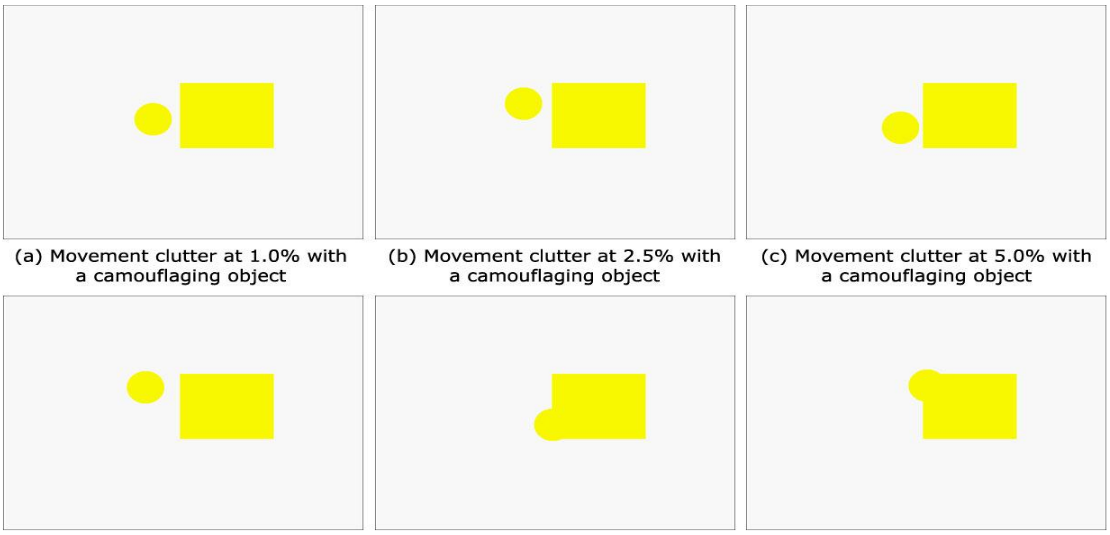
(c) Image clutter at 50% with a camouflaging object

(d) Image clutter at 70% with a camouflaging object

(e) Image clutter at 90% with a camouflaging object

(f) Image clutter at 100% with a camouflaging object

Figure C.10 Different percentage levels of image clutter during the occurrence of camouflage with the minimum value being 10% and maximum value being 100%.



(a) Movement clutter at 1.0% with a camouflaging object

(b) Movement clutter at 2.5% with a camouflaging object

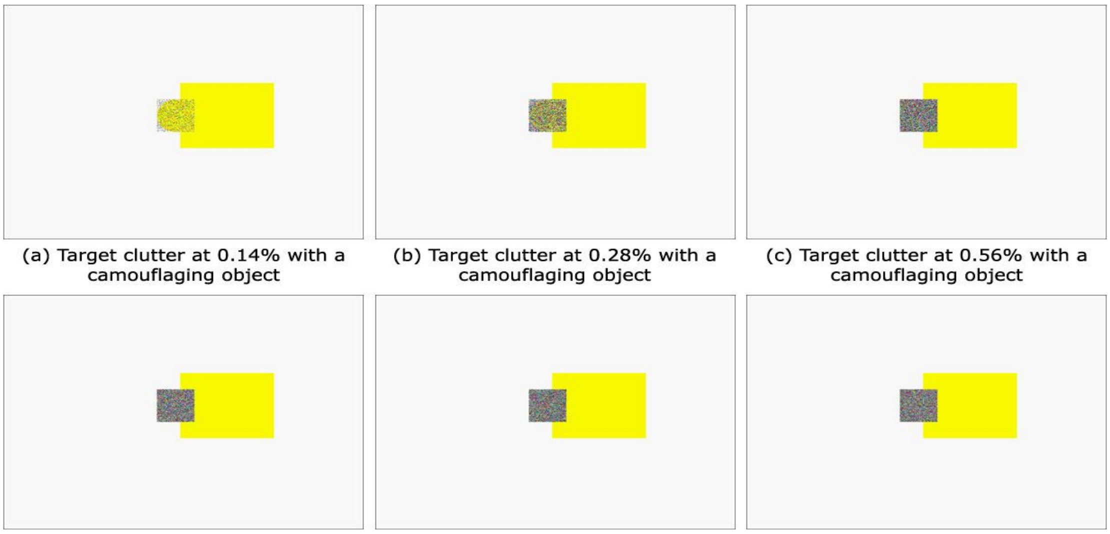
(c) Movement clutter at 5.0% with a camouflaging object

(d) Movement clutter at 7.5% with a camouflaging object

(e) Movement clutter at 9.0% with a camouflaging object

(f) Movement clutter at 10.0% with a camouflaging object

Figure C.11 Different percentage levels of motion clutter during the occurrence of camouflage with the minimum value being 1.0% and maximum value being 10.0%.



(a) Target clutter at 0.14% with a camouflaging object

(b) Target clutter at 0.28% with a camouflaging object

(c) Target clutter at 0.56% with a camouflaging object

(d) Target clutter at 0.98% with a camouflaging object

(e) Target clutter at 1.12% with a camouflaging object

(f) Target clutter at 1.26% with a camouflaging object

Figure C.12 Different percentage levels of target clutter during the occurrence of camouflage with the minimum value being 0.14% and maximum value being 1.26%.

References

Adam, A., Rivlin, E. & Shimshoni, I. (2006) Published. Robust Fragments-based Tracking using the Integral Histogram. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 17-22 June 2006. pp.798-805.

Ahad, A. R., Ogata, T., Tan, J. K., Kim, H. S. & Ishikawa, S. (2007) Published. Performance of Multi-directional MHI for Human Motion Recognition in the Presence of Outliers. *33rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2007).* 2007. pp.2366-2370.

Ahere, F. J., Thacker, N. A. & Rockett, P. I. (1997). The Bhattacharyya Metric as an Absolute Similiarity Measure for Frequency Coded Data. *Kybernetika,* 32 (4)**,** 1-7.

Arnaud, E. & Mémin, E. (2007). Partial Linear Gaussian Models for Tracking in Image Sequences Using Sequential Monte Carlo Methods. *International Journal of Computer Vision,* 74 (1)**,** 75-102.

Arulampalam, M. S., Maskell, S., Gordon, N. & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on,* 50 (2)**,** 174-188.

Babenko, B., Yang, M.-H. & Belongie, S. (2009) Visual Tracking with Online Multiple Instance Learning. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, USA, pp.983-990

Babu, R., Perez, P. & Bouthemy, P. (2006). Kernel-Based Robust Tracking for Objects Undergoing Occlusion. *Computer Vision – ACCV 2006.* pp.353-362.

Babu, R. V., Pérez, P. & Bouthemy, P. (2007). Robust tracking with motion estimation and local Kernel-based color modeling. *Image and Vision Computing,* 25 (8)**,** 1205-1216.

Babu, R. V. & Ramakrishnan, K. R. (2004). Recognition of human actions using motion history information extracted from the compressed video. *Image and Vision Computing,* 22 (8)**,** 597-607.

Bayes, T. & Price, R. (1763). An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. *Philosophical Transactions,* 53**,** 370-418.

Behera, R. K., Kharade, P., Yerva, S., Dhane, P., Jain, A. & Kutty, K. (2012) Published. Multi-camera based surveillance system. *Information and Communication Technologies (WICT), 2012 World Congress on*, Oct. 30 2012-Nov. 2 2012 2012. pp.102-108.

Bennett, B., Magee, D. R., Cohn, A. G. & Hogg, D. C. (2008). Enhanced tracking and recognition of moving objects by reasoning about spatio-temporal continuity. *Image and Vision Computing,* 26 (1)**,** 67-81.

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society,* 35**,** 99–109.

Bilmes, J. (1997) *A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*, Berkeley, California, Report number: TR-97-02.

Bishop, C., M. (1995). *Neural Networks for Pattern Recognition,* editionNew York, Oxford University Press, Inc.

Black, J., Ellis, T. & Rosin, P. (2003) Published. A Novel Method for Video Tracking Performance Evaluation. *The joint IEEE International Workshop on Visual*

Surveillance and Performance Evaluation of Tracking and Surveillance, October 11-12 2003 Nice, France. pp.125-132.

Bobick, A. & Davis, J. (1996a) Published. An appearance-based representation of action. *Proceedings of the 13th International Conference on Pattern Recognition* 1996a. pp.307-312.

Bobick, A. & Davis, J. (1996b) Published. Real-time recognition of activity using temporal templates. *Proceedings 3rd IEEE Workshop on Applications of Computer Vision (WACV '96)*, 1996b. pp.39-42.

Bobick, A. & Davis, J. (1997). Action Recognition Using Temporal Templates. *IN:* Shah, M. & Jain, R. (eds.) *Motion-based recognition.* Springer, pp.125-146.

Bobick, A., Intille, S., Davis, J., Baird, F., Pinhanez, C., Campbell, L., Ivanov, Y., Schutte, A. & Wilson, A. (1999). The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment. *Presence: Teleoperators and Virtual Environments,* 8 (4)**,** 369-393.

Bobick, A. F. & Davis, J. W. (2001). The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 23 (3)**,** 257-267.

Boehme, H. J., Braumann, U. D., Brakensiek, A., Corradini, A., Krabbes, M. & Gross, H. M. (1998) Published. User localisation for visually-based human-machine-interaction. *Proceedings of the third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998. pp.486-491.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing,* 34 (3)**,** 344-371.

Box, G. E. P. & Muller, M. E. (1958). A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics,* 29 (2)**,** 610-611.

Bradski, G. R. & Davis, J. (2000) Published. Motion segmentation and pose recognition with motion history gradients. *Fifth IEEE Workshop on Applications of Computer Vision*, 2000. pp.238-244.

Bradski, G. R. & Davis, J. W. (2002). Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications,* 13 (3)**,** 174-184.

Brasnett, P., Mihaylova, L., Bull, D. & Canagarajah, N. (2007). Sequential Monte Carlo tracking by fusing multiple cues in video sequences. *Image and Vision Computing,* 25 (8)**,** 1217-1227.

Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal,* 4 (1)**,** 25-30.

Bromiley, P. A., Thacker, N. A., Scott, M. L. J., Pokric, M., Lacey, A. J. & Cootes, T. F. (2003). Bayesian and non-Bayesian probabilistic models for medical image analysis. *Image and Vision Computing,* 21 (10)**,** 851-864.

Bullock, D. J. & Zelek, J. S. (2004). Real-time tracking for visual interface applications in cluttered and occluding situations. *Image and Vision Computing,* 22 (12)**,** 1083-1091.

Cappe, O., Godsill, S. J. & Moulines, E. (2007). An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proceedings of the IEEE,* 95 (5)**,** 899-924.

Castle, R. O., Klein, G. & Murray, D. W. (2011a). Wide-area augmented reality using camera tracking and mapping in multiple regions. *Computer Vision and Image Understanding,* 115 (6)**,** 854-867.

Castle, R. O. & Murray, D. W. (2011b). Keyframe-based recognition and localization during video-rate parallel tracking and mapping. *Image and Vision Computing,* 29 (8)**,** 524-532.

Chandesa, T., Pridmore, T. & Bargiela, A. (2009a) Published. Detecting occlusion and camouflage during visual tracking. *IEEE International Conference on Signal and Image Processing Applications (ICSIPA '09)*, November 18-19 2009a Kuala Lumpur, Malaysia. pp.468-473.

Chandesa, T., Pridmore, T. & Bargiela, A. (2009b) *Using Process-Behaviour charts to detect Occlusion and Camouflage in Visual Tracking*. Presented in: Research for a Better Tomorrow: Impacts in the 21st Centrury, University of Nottingham, Malaysia Campus.

Cheung, K.-W., Yeung, D.-Y. & Chin, R. T. (2002). Bidirectional Deformable Matching with Application to Handwritten Character Extraction. *IEEE Trans. Pattern Anal. Mach. Intell.,* 24 (8)**,** 1133-1139.

Chockalingam, P., Pradeep, N. & Birchfield, S. (2009) Published. Adaptive fragments-based tracking of non-rigid objects using level sets. *Computer Vision, 2009 IEEE 12th International Conference on*, Sept. 29 2009-Oct. 2 2009 2009. pp.1530-1537.

Cipra, B. (1999). *Engineers look to Kalman filtering for guidance* [Online]. Available: http://www.cs.unc.edu/~welch/kalman/siam_cipra.html [Accessed 3rd October 2006].

Collins, R. T., Liu, Y. & Leordeanu, M. (2005). Online Selection of Discriminative Tracking Features. *IEEE Trans. Pattern Anal. Mach. Intell.,* 27 (10)**,** 1631-1643.

Comaniciu, D. (2003). An Algorithm for Data-Driven Bandwidth Selection. *IEEE Trans. Pattern Anal. Mach. Intell.,* 25 (2)**,** 281-288.

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

Davis, J. (1999) *Recognizing Movement using Motion Histograms*, Cambridge, MA, Report number: #487.

Davis, J. & Bobick, A. (1997) Published. The Representation and Recognition of Action Using Temporal Templates. *IEEE Conference on Computer Vision and Pattern Recognition*, 1997. pp.928-934.

Davis, J. W. (2001) Published. Hierarchical motion history images for recognizing human motion. *Proceedings of the IEEE Workshop on Detection and Recognition of Events in Video*, July 8th 2001 Vancouver, Canada. pp.39-46.

Davis, J. W. & Bobick, A. F. (1998) Published. Virtual PAT: A virtual personal aerobics trainer. *Workshop on Perceptual User Interfaces (PUI'98)*, 1998. pp.13-18.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological),* 39 (1)**,** 1-38.

Djouadi, A., Snorrason, O. & Garber, F. D. (1990). The quality of training sample estimates of the Bhattacharyya coefficient. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 12 (1)**,** 92-97.

Dolgov, D. & Thrun, S. (2008) Published. Detection of principal directions in unknown environments for autonomous navigation. *Proceedings of the Robotics: Science and Systems IV (RSS-08)*, June 2008 Zurich, Switzerland.

Doucet, A., Godsill, S. & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing,* 10 (3)**,** 197-208.

Doucet, A. & Johansen, A. M. (2008). A Tutorial on Particle Filtering and Smoothing: Fifteen years Later. *IN:* Crisan, D. & Rozovsky, B. (eds.) *Oxford Handbook of Nonlinear Filtering.*

Ellis, T. (2002) Published. Performance Metrics and Methods for Tracking in Surveillance. *3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, June 1 2002 Copenhagen, Denmark. pp.26-31.

Ess, A., Leibe, B., Schindler, K. & L., V. G. (2009) Published. Moving obstacle detection in highly dynamic scenes. *International Conference on Robotics and Automation (ICRA'09)*, May 2009 Kobe, Japan.

Figueiredo, M. A. T. (2003). Adaptive Sparseness for Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.,* 25 (9)**,** 1150-1159.

Fisher, R., Perkins, S., Walker, A. & Wolfart, E. (2003). *Constrast Stretching* [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm [Accessed 21st January 2010].

Foley, J. D., van Dam, A., Feiner, S. K. & Hughes, J. F. (1995). *Computer Graphics: Principles and Practice in C,* 2 edition, Addison-Wesley.

Forsyth, D. A. & Ponce, J. (2003). *Computer Vision: A Modern Approach,* editionNew York Prentice Hall.

Fukunaga, K. & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on,* 21 (1)**,** 32-40.

Gonzalez, R. C. & Woods, R. E. (2002). *Digital Image Processing,* Second Edition editionNew Jersey, Prentice Hall.

Greenhill, D., Renno, J., Orwell, J. & Jones, G. A. (2008). Occlusion analysis: Learning and utilising depth maps in object tracking. *Image and Vision Computing,* 26 (3)**,** 430-441.

Greenspan, M., Limin, S. & Jasiobedzki, P. (2004) Published. Efficient tracking with the Bounded Hough Transform. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, June 27th - July 2nd 2004. pp.I-520-I-527.

Han, B. & Davis, L. (2005) Published. On-line density-based appearance modeling for object tracking. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 17-21 Oct. 2005 2005. pp.1492-1499 Vol. 2.

Hanzi, W., Suter, D., Schindler, K. & Chunhua, S. (2007). Adaptive Object Tracking Based on an Effective Appearance Filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 29 (9)**,** 1661-1667.

Haug, A. J. (2005). A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian process. The MITRE Corporation.

Hearn, D. & Baker, M. P. (2004). *Computer Graphics with OpenGL,* 3 editionUpper Saddle River, NJ, Pearson/Prentice Hall.

Holz, D., Lorken, C. & Surmann, H. (2008) Published. Continuous 3D sensing for navigation and SLAM in cluttered and dynamic environments. *Information Fusion, 2008 11th International Conference on*, 2008. pp.1-7.

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on,* 8 (2)**,** 179-187.

Isard, M. & Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. *Computer Vision (ECCV '96).* pp.343-356.

Isard, M. & Blake, A. (1998a). CONDENSATION—Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision,* 29 (1)**,** 5-28.

Isard, M. & Blake, A. (1998b) ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework. *Proceedings of the 5th European Conference on Computer Vision-Volume I - Volume I*, Springer-Verlag, pp.893-908

Isard, M. & Blake, A. (1998c) A Mixed-State Condensation Tracker with Automatic Model-Switching. *Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, pp.107-112

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

Jepson, A. D., Fleet, D. J. & El-Maraghi, T. R. (2001) Published. Robust online appearance models for visual tracking. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001 2001. pp.I-415-I-422 vol.1.

Jeyakar, J., Babu, R. V. & Ramakrishnan, K. R. (2008). Robust object tracking with background-weighted local kernels. *Computer Vision and Image Understanding,* 112 (3)**,** 296-309.

KaewTrakulPong, P. & Bowden, R. (2003). A real time adaptive visual surveillance system for tracking low-resolution colour targets in dynamically changing scenes. *Image and Vision Computing,* 21 (10)**,** 913-929.

Kailath, T. (1967). The Divergence and Bhattacharyya Distance Measures in Signal Selection. *Communication Technology, IEEE Transactions on,* 15 (1)**,** 52-60.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME of Journal of Basic Engineering,* (82 (Series D))**,** 35-45.

Karavasilis, V., Nikou, C. & Likas, A. (2011). Visual tracking using the Earth Mover's Distance between Gaussian mixtures and Kalman filtering. *Image and Vision Computing,* 29 (5)**,** 295-305.

Kelly, P. (2010). *Law's Texture Measure* [Online]. Available: http://www.ccs3.lanl.gov/~kelly/ZTRANSITION/notebook/laws.shtml [Accessed 20th January 2010].

Kim, H.-C. & Ghahramani, Z. (2006). Bayesian Gaussian Process Classification with the EM-EP Algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.,* 28 (12)**,** 1948-1959.

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics,* 5 (1)**,** 1-25.

Kotecha, J. H. & Djuric, P. M. (2003). Gaussian particle filtering. *Signal Processing, IEEE Transactions on,* 51 (10)**,** 2592-2601.

Kwak, S., Nam, W., Han, B. & Han, J. H. (2011) Published. Learning occlusion with likelihoods for visual tracking. *Computer Vision (ICCV), 2011 IEEE International Conference on*, 6-13 Nov. 2011. pp.1551-1558.

Kwon, J. & Lee, K. M. (2011) Tracking by Sampling Trackers. *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, pp.1195-1202

Lanz, O. (2006). Approximate Bayesian multibody tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 28 (9)**,** 1436-1449.

Lascio, R. D., Foggia, P., Percannella, G., Saggese, A. & Vento, M. (2013). A real time algorithm for people tracking using contextual reasoning. *Computer Vision and Image Understanding,* (0).

Laws, K. I. (1979) Published. Texure Energy Measure. *In DARPA Image Understanding Workshop*, 1979 Los Angeles. pp.47-51.

Laws, K. I. (1980a) Published. Rapid Texture Identification. *Proc. SPIE Conf. Image Processing for Missile Guidance*, 1980a. pp.376-380.

Laws, K. I. (1980b). *Textured Image Segmentation.* PhD Dissertation, University of Southern California.

Leichter, I. D. O., Lindenbaum, M. & Rivlin, E. (2006). A General Framework for Combining Visual Trackers – The "Black Boxes" Approach. *International Journal of Computer Vision,* 67 (3), 343-363.

Li, P., Zhang, T. & Pece, A. E. C. (2003). Visual contour tracking based on particle filters. *Image and Vision Computing,* 21 (1), 111-123.

Lim, C.-P., Leong, J.-H. & Kuan, M.-M. (2005). A Hybrid Neural Network System for Pattern Classification Tasks with Missing Features. *IEEE Trans. Pattern Anal. Mach. Intell.,* 27 (4), 648-653.

Ma, Y., Yu, Q. & Cohen, I. (2009). Target tracking with incomplete detection. *Computer Vision and Image Understanding,* 113 (4), 580-587.

MacQueen, J. (1967) Published. Some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, 1967. Univ. of Calif. Press, pp.281-297.

Maggio, E. & Cavallaro, A. (2009). Accurate appearance-based Bayesian tracking for maneuvering targets. *Computer Vision and Image Understanding,* 113 (4), 544-555.

Maxwell, B. A. & Brubaker, S. J. (2003) Published. Texture Edge Detection Using the Compass Operator. *Proceedings of British Machine Vision Conference*, September 2003 Norwich, UK. pp.549-558.

Milstein, A., Sánchez, J. N. & Williamson, E. T. (2002) Robust global localization using clustered particle filtering. *Eighteenth national conference on Artificial intelligence*, Edmonton, Alberta, Canada,American Association for Artificial Intelligence, pp.581-586

Mittal, A. & Davis, L. S. (2003). M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. *International Journal of Computer Vision,* 51 (3), 189-203.

Morris, D., Hoffman, R. & McLean, S. (2008) Published. Ladar-based vehicle detection and tracking in cluttered environments. *26th Army Science Conference*, December 2008 Orlando.

Nagenborn, R. (2003). *Robot localization and Kalman filters.* M.Sc Master of Science, Institute of Information and Computing Science, Utrecht University.

Nelson, L. S. (1984). Technical Aids. *Journal of Quality Technology,* 16 (4), 238-239.

Nguyen, H. T. & Smeulders, A. W. M. (2004). Fast occluded object tracking by a robust appearance filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 26 (8), 1099-1104.

Nummiaro, K., Koller-Meier, E. & Van Gool, L. (2003). An adaptive color-based particle filter. *Image and Vision Computing,* 21 (1), 99-110.

Pérez, P. & Vermaak, J. (2005) Published. Bayesian tracking with auxiliary discrete processes. Application to detection and tracking of objects with occlusions. *In Proc. ICCV'05 Workshop on Dynamical Vision*, October 2005 Beijing, China.

Pitt, M. & Shephard, N. (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association,* 94 (446), 590-599.

Pound, M., Naeem, A., French, A. & Pridmore, T. (2007) Published. Quantitative and Qualitative Evaluation of Visual Tracking Algorithms Using Statistical Tests. *10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, October 14 2007 Rio de Janeiro, Brazil. pp.33-40.

Rabiner, L. & Juang, B. H. (1993). *Fundamentals of speech recognition,* editionEnglewood Cliffs, NJ, Prentice-Hall, Inc.

Randen, T. & Husoy, J. H. (1999). Filtering for texture classification: a comparative study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 21 (4)**,** 291-310.

Redner, R. A. & Walker, H. F. (1984). Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review,* 26 (2)**,** 195-239.

Roberts, S. W. (1959). Control Chart Test based on Geometric Moving Averages. *Technometrics,* 1**,** 239-250.

Rosales, R. (1998) *Recognition of Human Action Using Moment-Based Features*, Boston, MA, Report number: 1998-020

Rosales, R. & Sclaroff, S. (1998) Published. Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modelling. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Workshop on the Interpretation of Visual Motion*, 1998 Santa Barbara.

Ross, D., Lim, J. & Yang, M. (2004) Adaptive Probabilistic Visual Tracking with Incremental Subspace Update. In: Pajdla, T. & Matas, J., (eds.) *Proc. Eighth European Conference on Computer Vision (ECCV 2004)*, Springer, pp.470-482

Ruzon, M. A. & Tomasi, C. (1999) Color edge detection with the compass operator. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.160-166

Senior, A., Hampapur, A., Tian, Y.-L., Brown, L., Pankanti, S. & Bolle, R. (2006). Appearance models for occlusion handling. *Image and Vision Computing,* 24 (11)**,** 1233-1243.

Shan, C., Tan, T. & Wei, Y. (2007). Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recognition,* 40 (7)**,** 1958-1970.

Shen, J., Yang, W., Lu, Z. & Liao, Q. (2012). Information integration for accurate foreground segmentation in complex scenes. *Image Processing, IET,* 6 (5)**,** 596-605.

Shewhart, W. A. (1931). *The ecomomic control of the quality of manufactured product,* editionNew York, Macmillan.

Sim, R., Elinas, P. & Little, J. (2007). A Study of the Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-Based SLAM. *International Journal of Computer Vision,* 74 (3)**,** 303-318.

Simon, D. (2001) Kalman Filtering, 14 (6), 1 June, pp.72-79.

Song, B., Jeng, T.-Y., Staudt, E. & Chowdhury, A. K. R. (2010) A stochastic graph evolution framework for robust multi-target tracking. *Proceedings of the 11th European conference on Computer vision (I)*, Heraklion, Crete, Greece, pp.605-619

Steinhaus, H. (1956). Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci,* 1**,** 801-804.

Stolkin, R., Rees, D., Talha, M. & Florescu, I. (2012) Published. Bayesian fusion of thermal and visible spectra camera data for region based tracking with rapid background adaptation. *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, 13-15 Sept. 2012 2012. pp.192-199.

Talha, M. & Stolkin, R. (2012) Published. Adaptive fusion of infra-red and visible spectra camera data for particle filter tracking of moving targets. *Sensors, 2012 IEEE*, 28-31 Oct. 2012 2012. pp.1-4.

Town, C. (2007). Multi-sensory and Multi-modal Fusion for Sentient Computing. *International Journal of Computer Vision,* 71 (2)**,** 235-253.

Trucco, E. & Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision,* editionUpper Saddle River, NJ Prentice Hall.

Ventura, J. & Hollerer, T. (2012) Published. Wide-area scene mapping for mobile visual tracking. *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, 5-8 Nov. 2012 2012. pp.3-12.

Verma, R. C., Schmid, C. & Mikolajczyk, K. (2003). Face detection and tracking in a video by propagating detection probabilities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 25 (10)**,** 1215-1228.

Vermaak, J., Doucet, A. & Perez, P. (2003) Published. Maintaining multimodality through mixture tracking. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, October 13-16 2003. IEEE Computer Society, pp.1110-1116.

Weinland, D., Ronfard, R. & Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding,* 104 (2-3)**,** 249-257.

Welch, G. & Bishop, G. (1995). *An introduction to the Kalman filter.* Technical Report - TR95-041, Department of Computer Science, University of North Carolina at Chapel Hill.

Williams, D., Liao, X., Xue, Y., Carin, L. & Krishnapuram, B. (2007). On Classification with Incomplete Data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 29 (3)**,** 427-436.

Xiang, X. (2011) Published. A brief review on visual tracking methods. *Intelligent Visual Surveillance (IVS), 2011 Third Chinese Conference on*, 1-2 Dec. 2011 2011. pp.41-44.

Xu, L. & Michael, I. J. (1996). On convergence properties of the em algorithm for gaussian mixtures. *Neural Comput.,* 8 (1)**,** 129-151.

Yang, H., Shao, L., Zheng, F., Wang, L. & Song, Z. (2011). Recent advances and trends in visual tracking: A review. *Neurocomput.,* 74 (18)**,** 3823-3831.

Yilmaz, A., Javed, O. & Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv.,* 38 (4)**,** 13.

Zhang, J., Li, Z. & Wang, Z. (2009). Control chart based on likelihood ratio for monitoring linear profiles. *Computational Statistics & Data Analysis,* 53 (4)**,** 1440-1448.

Zhang, Z., Hu, Y., Chan, S. & Chia, L.-T. (2008). Motion Context: A New Representation for Human Action Recognition. *Computer Vision – ECCV 2008.* Springer Berlin / Heidelberg, pp.817-829.

Zhou, L., Kaiqi, H. & Tieniu, T. (2012). Foreground Object Detection Using Top-Down Information Based on EM Framework. *Image Processing, IEEE Transactions on,* 21 (9)**,** 4204-4217.

Ziyan, U., Sabuncu, M., Grimson, W. & Westin, C.-F. (2009). Consistency Clustering: A Robust Algorithm for Group-wise Registration, Segmentation and Automatic Atlas Construction in Diffusion MRI. *International Journal of Computer Vision*.