MULTIPLE OBJECT TRACKING USING PARTICLE FILTERS

by

HWANGRYOL RYU

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2006

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Manfred Huber, for endless excellent guidance in exploring different aspects of the problem. I would like to thank my committee members, Dr. Jean Gao and Dr. Gergely Zaruba for encouraging and giving invaluable advice. I also thank Dr. Christine Murray for taking the time to review my dissertation.

I would like to thank my friends, Changhoon Lee, Kyungseo Park, Byoungyong Lee, Chulho Ahn, and all other member of the Robotics lab.

A special thanks goes to my parents, my brother and friends, who have given me invaluable supports financially and spiritually. Finally, I would like to thank my beloved wife, Hyunkyung Kim and my son, Roy Jisung Ryu for their patience and love.

May 8, 2006

ABSTRACT

MULTIPLE OBJECT TRACKING USING PARTICLE FILTERS

Publication No. _____

Hwangryol Ryu, MS

The University of Texas at Arlington, 2006

Supervising Professor: Manfred Huber

We describe a novel extension to the Particle Filter algorithm for tracking multiple objects. The recently proposed algorithms and the variants for multiple object tacking estimate multi-modal posterior distributions that potentially represent the multiple peaks (i.e., multiple tracked objects). However, the specific state representation does not demonstrate creation, deletion, and more importantly partial/complete occlusion of the objects. Furthermore, the weakness of the Particle Filter such that the representation may increasingly bias the posterior density estimates toward objects with dominant likelihood makes the multiple object tracking algorithms more difficult. To circumvent a sample depletion problem and maintain the computational complexity as good as the *mixture* Particle filters under certain assumptions - (1) targets move independently, (2) targets are not transparent, (3) each pixel of the image can only come from one of the targets - we proposed a new approach dealing with partial and complete occlusions of a fixed number of objects in an efficient manner that provides a robust means of tracking each object by projecting particles into the image space and back to a particle space. The projection procedure between the particle space and the image space represents an

important probability density function not only to give more trust to a target being visible, but also to explain an occluded target. In addition, while *joint* Particle filters suffer from the curse of dimensionality in the number of the targets being tracked, the proposed algorithm only adds a constant factor to the computational cost of the standard Particle filters.

To present qualitative results, experiments were performed using color-based tracking of multiple rectangular boxes of different colors. The experiments demonstrated that the Particle filters implemented using the proposed method effectively and precisely track multiple targets, whereas the standard Particle filters failed to track the multiple targets.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT iv

LIST OF FIGURES viii

LIST OF TABLES x

Chapter

1. INTRODUCTION 1

    1.1 Motivation 1

    1.2 Challenges and Contributions 2

    1.3 Organization 3

2. FROM BAYESIAN FILTER TO PARTICLE FILTER 5

    2.1 Introduction 5

        2.1.1 From Filtering Theory To Particle Filter 5

    2.2 Bayesian Filter 7

        2.2.1 Recursive Bayesian Estimation 8

    2.3 Particle Filter 10

        2.3.1 Importance Sampling (IS) 11

        2.3.2 Sequential Importance Sampling (SIS) 12

        2.3.3 Sampling Importance Resampling (SIR) 14

3. RELATED WORK 17

    3.1 Multiple Instantiations of Single Target Tracker 17

        3.1.1 State Space Extension 18

4. DISTRIBUTED MULTI-TARGET PARTICLE FILTER 21

4.1 Introduction 21
4.2 Multi-Target Filtering Distributions 23
4.2.1 Target Filtering Distribution 24
4.2.2 Joint Observation Likelihood Model 25
4.2.3 Filter Deletion 31
4.2.4 Filter Creation 32
4.3 Computational Complexity 33
4.4 SIR Multi-Target Particle Filter 35
5. EXPERIMENTAL RESULTS 37
5.1 State Space Model 37
5.2 Observation Likelihood Model 38
5.2.1 Color Likelihood Model 38
5.2.2 Compute Expected Observation Model 40
5.3 Experimental Results 41
5.3.1 Standard Particle Filters 42
5.3.2 Distributed Multi-Target Particle Filter 51
6. CONCLUSIONS 75
REFERENCES 77
BIOGRAPHICAL STATEMENT 80

LIST OF FIGURES

Figure Page

4.1 Given the particle weight, each particle weight is projected into the image space evenly according to the size of the filter 29

4.2 Projection step between the particle space and the image space 35

5.1 Mean of errors in Cartesian distance and error bars using standard Particle filter on target 1 (a) and target 2 (b) 45

5.2 Sequence of the images demonstrates the sample depletion problem using standard Particle filter with 1000 particles 46

5.3 Sequence of the images demonstrates the sample depletion problem using standard Particle filter with 1000 particles 47

5.4 Target disappearance experiment using standard Particle filter with 1000 particles 48

5.5 Target disappearance experiment using standard Particle filter with 1000 particles 49

5.6 Target disappearance experiment using standard Particle filter with 1000 particles 50

5.7 (a) and (b) are mean errors in Cartesian distance and error bars using multi-target Particle filter on target 1 and target 2, respectively 52

5.8 Tracking two targets in the X axis 53

5.9 Tracking two targets in the Y axis 54

5.10 (a) and (b) are images before occlusion and (c) and (d) are visibility and invisibility density for targets at frame 84 55

5.11 (a) and (b) are images while occlusion and (c) and (d) are visibility and invisibility density for targets at frame 91 56

5.12 (a) and (b) are images while occlusion (c) and (d) are visibility and invisibility density for target at frame 189 57

5.13 (a) and (b) are images after occlusion (c) and (d) are visibility and invisi-

bility density for target at frame 201 58

5.14 Mean of error in Cartesian distance between estimated state and true state and error bars using distributed multi-target Particle filter 60

5.15 Tracking two targets in the X axis 61

5.16 Tracking two targets in the Y axis 62

5.17 (a) and (b) Images at frame 83. (c), (d), and (e) are visibility and invisibility density of green, red, and blue target, respectively 63

5.18 (a) and (b) Images at frame 117. (c), (d), and (e) are visibility and invisibility density of green, red, and blue target, respectively 64

5.19 (a) and (b) Images at frame 130. (c), (d), and (e) are visibility and invisibility density of green, red, and blue target, respectively 65

5.20 (a) and (b) Images at frame 143. (c), (d), and (e) are visibility and invisibility density of green, red, and blue target, respectively 66

5.21 (a) and (b) Images at frame 184. (c), (d), and (e) are visibility and invisibility density of green, red, and blue target, respectively 67

5.22 Importance Weight Distribution for three targets 69

5.23 Visual tracking results on target deletion 70

5.24 Visual tracking results on target creation 71

5.25 (a) and (b) Mean of errors in Cartesian distance and error bars using the standard Particle filter and multi-target Particle filter, respectively 73

5.26 Mean of errors in Cartesian distance and error bars using standard Particle filter 74

LIST OF TABLES

Table Page

2.1 Filtering via SIS 14

2.2 Resampling Algorithm 15

2.3 SIR Particle Filter 16

CHAPTER 1

INTRODUCTION

1.1 Motivation

Robotic and embedded systems have become increasingly pervasive, complex, and demanded in every-day applications, ranging from life support systems to personal computing devices. In particular, developing the technologies for human robot interaction gives tremendous benefits to our daily life. For example, intelligent robots can perform potentially dangerous work, delivery services, or trivial work which people do not like to do. Service robots could do household chores at home such as cleaning or washing clothes. Since nursing robots were first developed to help the physically handicapped in 1986, many different types of nursing robot platforms have been invented for the elderly and disabled that are capable of performing various tasks. Such devices and systems will return a measure of independence to many bedridden persons as well as reducing the number of those in need of hospitalization and constant attendance. These technologies can improve the quality of life by increasing the amount of freedom and the range of possibilities.

How can we make this possible? From the programmers' perspectives, is it possible to make robots smart enough to learn every single task that happens in our daily life? How do we know what tasks are needed or not? Our motivations come from these questions. We hope that autonomous and automated robots could be a solution performing these tasks in the near future. In order for autonomous robots to program themselves, they should utilize a vision system to interpret perceptual information because a camera

provides tremendous information that would not be available from other sensory devices such as ultrasonic ranger, light detecting sensor, etc. With the visual information the robots should identify and track objects that are of interest to its task. For this, robust real-time multi-target tracking methods have to be developed and the rest of this thesis will discuss a novel technology for this task in detail.

1.2 Challenges and Contributions

Real-time target tracking is a critical task in many computer vision applications such as robotic visual servoing tasks, surveillance, gesture recognition for human-machine interfaces and smart environments, augmented reality and visual effects, motion capture, medical and meteorological imaging, driver assistance, etc. Even though tasks of each application may be different in the visual domain, the tracking approaches generally fall into one of two groups: object recognition and object localization or tracking. Object recognition has been widely researched and many commercial vision systems have been successfully developed and deployed. What interests me the most is how to track and localize a set of targets in a real-time process, while maintaining the correct target recognition.

Michael Isard and Andrew Black [1] first proposed a stochastic framework for tracking curves in visual clutter, using a sampling algorithm, called "CONDENSATION" in the year 1998. Originally this idea was rooted in statistics, control theory, and computer vision, but thanks to the rapid increase of computational power and the availability of cheap memory, it was possible to run this sampling algorithm and an object recognition algorithm at the same time. This sampling algorithm has many different names such as Monte Carlo Simulation, Particle filter,Bootstrap filtering, interacting particle approximation, sequential Monte Carlo methods, SIS, SIR, ASIR, RPF, and so on. Chapter 2

will explain these algorithms in detail. One of the common properties of these sampling algorithms is that they are designed to track one target for each filter if they are not a *joint* filter. However, our thesis extends this property to multiple targets.

Maintaining multiple targets in a Particle filter, which is originally used to represent a single target, is a challenging task. In particular, sample depletion, which will be explained in Chapter 4, happens when one target has a stronger measurement support than the other target. Also, the multi-modality that may arise in the single state distribution due to the multiple targets or clutter makes it hard to estimate the state. The standard filtering distribution contains usually one peak representing the potential target state. So we investigate a possible solutions to these challenges as follows: (1) an idea similar to the *mixture* Particle filter is used to maintain each target distribution, (2) the standard observation likelihood model is factored into two terms to represent target occlusion situations. The proposed approach assumes that each target evolves independently, but in the case that targets occlude each other, it allows each target to interact with the others while maintaining each target distribution. With a focus on tracking multiple targets, this approach attempts to approximate a joint Particle filter to interpret occlusion situations while distributing the target across separate Particle filters.

1.3 Organization

This section describes the organization of this thesis. Chapter 2 introduce filtering algorithms from Bayesian filters to Particle filters. Chapter 3 briefly reviews related work using the Particle filters in the context of multi-target tracking. Chapter 4 explains the main contributions of this thesis. The mathematical formulations are derived to explain the occlusion situation on the observation model. It also shows an implementation based on the derived formulations. Chapter 5 provides the results of the experiments

including the state space, the observation model, and the dynamic model used in the implementation. The experiments deal with real world problems involving the tracking of multiple rectangular color boxes in a video sequence. At last, this thesis is concluded in Chapter 6.

CHAPTER 2

FROM BAYESIAN FILTER TO PARTICLE FILTER

2.1 Introduction

Target tracking techniques are important elements of surveillance, guidance, or obstacle avoidance systems. Such devices are used to determine the number, position and movement of targets. In particular, to develop adaptive capabilities that interpret the perceptual information as described in the previous chapter, target tracking plays an important role in deriving functional representations of an observed task because it may contain a sequence of related tasks, which may be meaningful. In order to track these sequences, recursive target state estimation, *filtering*, has been used in many applications. This chapter discusses state estimation techniques in the context of target tracking.

This chapter touches on three major scientific areas: stochastic filtering theory, Bayesian theory, and Monte Carlo methods. These methods are closely related to the subject of this thesis - Particle filtering - but explaining the theoretical concept of each theory is beyond this thesis. Therefore, in the following sections only a brief description of each theory will be given.

2.1.1 From Filtering Theory To Particle Filter

Stochastic filtering theory was first established in the early 1940s by Norbert Wiener [2], [3] and Andrey N. Kolmogorov [4] and in 1960 the classical Kalman filer (**KF**) was published and has become the most popular filter as an optimal recursive Bayesian estimator to a somewhat restricted class of linear Gaussian problems for more than 40 years.

Recently there has been a surge of interest in nonlinear and non-Gaussian filtering due to the fast increase in computing power. Since the Kalman filter is limited by its assumptions, numerous nonlinear filtering methods along this line have been proposed and developed to overcome its limitation. The British researcher, Thomas Bayes, originally developed Bayesian theory in a posthumous publication in 1763 [5]. However, Bayes theory has not gained a lot of attention in the early days until Bayesian inference was developed by the French mathematician Pierre Simon De Laplace [6]. Bayesian inference, which applies Bayesian statistics to statistical inference, has become one of the important branches in statistics and has been successfully applied in statistical decision, detection and estimation, pattern recognition, and machine learning. In 1998 the first Particle filter for tracking curves in dense visual clutter was proposed by Michael Isard [1] and Andrew Blake [7].

The early idea of Monte Carlo was designed to estimate the number π, but the modern formulation of Monte Carlo methods started from physics in the 1940s and came to statistics later in the 1950s. Roughly speaking, a Monte Carlo technique is a sampling approach concentrating on a solution of complex systems which are analytically intractable. The Kalman filter was not suited to evaluate complex systems analytically and from the mid 1960s, the extended Kalman filter and the Gaussian sum filter were developed to solve the non-linear and non-Gaussian systems. Possibly due to the severe computational limitations of the time, the Monte Carlo methods have not gained huge attentions until recently. In the late 1980s, thanks to increasing computational power, sequential Monte Carlo approaches have proven their capability for approximating optimal solutions of non-linear and Gaussian systems with many successful applications in statistics [8], signal processing [3], machine learning, econometrics, automatics control, tracking ([9], [10]), communications, biology, and many others [11]. A number of tech-

niques for this type of filtering have been proposed and successfully applied, but our main focus is the sequential Monte Carlo estimation, collectively referred to as *particle filters*.

The following sections present a general review of stochastic filtering theory from a Bayesian perspectives: the Bayesian filter and Particle filter. To our interest, we derive the Bayesian distribution (i.e. *a posterior* distribution function) to have a mathematically tractable representation of non-Linear and non-Gaussian systems, often called the *state-space model*. After the Bayesian filter, we show the mathematical formulations of a recursive Bayesian filter using the Monte Carlo method.

2.2 Bayesian Filter

Filtering is an operation extracting quantitative information at time t by using data measured up to and including time t. The problem is to estimate sequentially the state of a dynamic system using a sequence of noisy measurements made of the system. We adopt the sate-space form to modeling dynamic systems. Now, to define the problem of nonlinear filtering, let us consider the following generic stochastic filtering problem in a dynamic state-space form:

$$\begin{aligned} \dot{x} &= f(t, x_t, u_t, d_t) \\ z_t &= g(t, x_t, u_t, v_t) \end{aligned} \tag{2.1}$$

here $\dot{x}$ and z_t are called the state equation and the measurement equation, respectively. x_t represents the sate vector $x_t \in \mathbb{R}^{n_x}$, where n_x is the dimension of the state vector, $\mathbb{R}$ is a set of real numbers and $t \in \mathbb{N}$ is the time index. It contains all relevant information required to describe the system under investigation. z_t is the measurement vector that

is related to the state vector, u_t is the system input vector, and d_t and v_t are the process (transition) noise and measurement noise respectively. In practice, we focus on the discrete time formulation with no external system input. In this context, the discrete filtering problem is defined as follow:

$$\begin{aligned} x_{t+1} &= f(x_t, d_t) \\ z_t &= g(x_t, v_t) \end{aligned} \tag{2.2}$$

where the first equation characterizes the state transition probability $p(x_{t+1}|x_t)$, whereas the second equation describes the measurement model $p(z_t|x_t)$. The probabilistic forms of the transition model and measurement model are ideally suited for the Bayesian approach, thus resulting in a rigorous general framework for dynamic state estimation problems.

2.2.1 Recursive Bayesian Estimation

We show the detailed derivation of recursive Bayesian estimation. As defined in the preceding section, we seek the filtered estimate of x_k based on the sequence of all available measurements $Z^t = z_1, ..., z_t$ up to time t. Since the problem is to recursively estimate the state x_t at time t, given the history of the observation data Z^t up to time t, it is required to construct the posterior probability density function (*pdf*) $p(x_t|Z^t)$. In principle, the pdf $p(x_t|Z^t)$ is obtained by the following two steps: prediction and update step. The prediction step is an *a priori* form of estimation, which is to derive information about what the quantity of interest will be like in the future and the update step is an *a posteriori* form of estimation in that the measurement at the current time t is used for estimation. We assume that the states follow a first-order Markov process and the

observations are independent of the given states. Let $p(x_t|Z^t)$ denote the conditional *pdf* of x_t. From Bayes law we can derive

$$\begin{aligned} p(x_t|Z^t) &= \frac{p(Z^t|x_t)p(x_t)}{p(Z^t)} \\ &= \frac{p(z_t, Z^{t-1}|x_t)p(x_t)}{p(z_t, Z^{t-1})} \\ &= \frac{p(z_t|Z^{t-1}, x_t)p(Z^{t-1}|x_t)p(z_t)}{p(z_t|Z^{t-1})p(Z^{t-1})} \\ &= \frac{p(z_t|Z^{t-1}, x_t)p(x_t|Z^{t-1})p(Z^{t-1})p(x_t)}{p(z_t|Z^{t-1})p(Z^{t-1})p(x_t)} \\ &= \frac{p(z_t|x_t)p(x_t|Z^{t-1})}{p(z_t|Z^{t-1})} \end{aligned} \tag{2.3}$$

The posterior density $p(x_t|Z^t)$ in Equation 2.3 is composed of three terms: *prior*, likelihood and evidence. The first term, $p(x_t|Z^{t-1})$, involves using the state transition model defined in Equation 2.2 to obtain the prediction density of the sate at time t via the Chapman-Kolmogorov equation [12]:

$$p(x_t|Z^{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|Z^{t-1})\,dx_{t-1} \tag{2.4}$$

where $p(x_t|x_{t-1})$ is the transition density of the state. The second term, $p(z_t|x_t)$ returns the measurement support with the noisy data. The last term, $p(z_t|Z^{t-1})$, involves an integral

$$p(z_t|Z^{t-1}) = \int p(z_t|x_t)p(x_t|Z^{t-1})\,dx_t \tag{2.5}$$

Suppose that the required *pdf* $p(x_{t-1}|Z^{t-1})$ at time t-1 is available. The prediction step in Equation 2.4 is carried out using the transition probability. At time step t when measurement z_t is available, the update step is then carried out. It involves the process

of Equation 2.3.

In general, the mathematical formulation of the recursive Bayesian estimation is only a conceptual solution in the sense that it cannot be solved analytically. Since it requires the storage of the entire *pdf* for implementation of the conceptual solution, the analytic solution of Equation 2.4 and 2.3 is intractable. Therefore, we should use an approximate or suboptimal Bayesian method which makes assumptions about the nature of the pdf (e.g., Kalman Filters) or which approximates a general pdf. The following section presents an introduction to an approximate solution for the estimation of recursive Bayesian filters that allows for general probability density functions (*pdf*).

2.3 Particle Filter

Particle filters are suboptimal filters. They perform sequential Monte Carlo (**SMC**) estimation based on a point mass representation of probability densities [13, 10, 14, 1, 15, 7, 12, 16, 17, 9, 18, 19, 20, 21]. Back in the 1950s the basic **SMC** framework in the form of sequential importance sampling had been introduced, but due to the lack of computational power, this idea had not largely gained attention until the 1970s. With the increasing computational power, **SMC** culminated in recursive state estimation methods. The major contribution to the development of the **SMC** method was the fast computation of the resampling step. As we shall describe later, the early implementation of Sequential Importance Sampling (**SIS**) suffers from sample depletion, which means that a point mass (i.e., particle) degenerates over time. The inclusion of the resampling step solved the degeneracy problem and made particle filters practical. The following subsections give a brief introduction of the three Monte Carlo sampling methods: importance sampling (**IS**), sequential importance sampling (**SIS**), and sampling importance resampling (**SIR**).

2.3.1 Importance Sampling (IS)

Monte Carlo methods use statistical sampling and estimation techniques to evaluate the solutions to mathematical problems. Let $\mathbf{X}^t = x_1, ..., x_t$ be the sequence of states up to time t. We denote $p(X^t|Z^t)$ to be the joint *pdf*, whose marginal is $p(x_t|Z^t)$. Let $\{X_t^i, \pi_t^i\}_{i=1}^N$, where N is the number of samples denote a random measure that characterizes the joint posterior $p(X_t^i|Z^t)$ where $\{X_t^i\}_{i=1}^N$ is a set of support points (i.e., particles) with associated weights $\{\pi_t^i\}_{i=1}^N$. The weights are normalized such that $\sum_i^N \pi_t^i = 1$. Then the joint *pdf* at time step t can be approximated as follows [11]:

$$p(X^t|Z^t) \approx \sum_{i=1}^{N} \pi_t^i \delta(X^t - X_t^i) \tag{2.6}$$

where $\delta(\cdot)$ is a Dirac-delta function with $\delta(X^t - X_t^i) = 0$ for all $X^t \neq X_t^i$ and 1 otherwise.

The main idea of importance sampling (**IS**) is to randomly pick a number of independent samples (N_p) from the distribution in the relatively important regions of the whole region in order to achieve computational efficiency. In a high dimensional space where the data is usually sparse and the region of interest where the target is located is a small region, this method is used to retrieve the samples from those regions. However, it is often impossible to create the true probability distribution $p(x)$ representing the state. Therefore, a proposal distribution $q(x)$ is used instead. The support of $q(x)$ is assumed to cover that of $p(x)$. This principle relies on the fact that if $q(\cdot)$ is not close to $p(\cdot)$, it can be imagined that the weights are very uneven, thus many samples are almost useless because of their negligible contributions. In a high dimensional space, the importance sampling estimate is likely dominated by a few samples with large importance weights. Consequently, a weighted approximation to the density $p(x)$ is given by

$$p(x) \approx \sum_{i=1}^{N} \pi^i \delta(x - x^i) \tag{2.7}$$

where $\pi^i \propto \frac{p(x^i)}{q(x^i)}$ is the normalized weight of the i^{th} particle [11].

2.3.2 Sequential Importance Sampling (SIS)

The sequential importance sampling algorithm forms the mathematical basis for most sequential Monte Carlo filters. This sequential Monte Carlo approach is known as bootstrap filtering [17], condensation algorithm [13], particle filter [11], and survival of the fittest [11]. The **SMC** is a technique for implementing a recursive Bayesian filter by Monte Carlo simulations. The main idea is to represent a posterior density function (*pdf*) by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples increases, the posterior *pdf* provides an increasingly accurate representation of the analytically intractable Bayesian estimator.

The sequential importance sampling (**SIS**) method essentially approximates the state of the system with randomly generated samples from some distribution. Therefore, choosing a good proposal distribution (i.e., importance density) is important to improve the efficiency of this method. How do we choose the good proposal distribution $q(x)$? It is usually difficult to find a good proposal distribution especially in a high-dimensional space. The basic idea of sequential importance sampling (**SIS**) comes from the above question and addresses it by deriving the proposal distribution sequentially, whereas importance sampling (**IS**) is not using sequential sampling. In the sequential case, the following equation shows the proposal distribution in a factorized form [11] so that importance sampling can be performed iteratively.

$$q(X^t|Z^t) = q(x_0)\prod_{i=1}^{t} q(x_i|X^{t-1}, Z^t) \tag{2.8}$$

An interested reader is referred to [22, 8] for derivation detail. The weight update equation is give by

$$\pi_t^i \propto \pi_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, z_t)} \tag{2.9}$$

and the posterior filtered density $p(x_t|Z^t)$ can be approximated as

$$q(x_t|Z^t) \approx \sum_{i=1}^{N} \pi_t^i \delta(x_t - x_t^i) \tag{2.10}$$

where the weights are defined in Equation 2.9. It can be shown that as $N \to \infty$, then approximation Equation 2.10 approaches the true posterior density $p(x_t|Z^t)$.

Recursive estimation filtering using sequential importance sampling thus consists of recursive propagation of importance weights (i.e., particle weights) π_t^i and support points x_t^i as each measurement is received sequentially. A description of **SIS** is summarized in Table 2.1 from [11].

2.3.2.1 Degeneracy Problem

The distribution of the important weights corresponding to the support points (i.e., particles) becomes the importance density function. It is shown in [11] that the unconditional variance of the importance weights increases over time, which is the so-called weight degeneracy problem. In practical terms, this problem means that after a few iterations of the algorithm, all but a few particles will have negligible normalized weights. This is disadvantageous since a lot of computing effort is wasted to update

Table 2.1. Filtering via SIS

$[x_t^i, \pi_{t\,i=1}^{i\,N}]$ = SIS $[x_{t-1}^i, \pi_{t-1\,i=1}^{i\quad N}, z_t]$
- FOR i = 1 : N
  - Draw $x_t^i \sim q(x_t|x_{t-1}^i, z_t)$
  - Assign the particle weight, π_t^i, according to Equation 2.9
- END FOR
- Calculate total weight: $t = SUM[\pi_{t\,i=1}^{i\,N}]$
- FOR i = 1 : N
  - Normalize: $\pi_t^i = t^{-1}\pi_t^i$
- END FOR

---

those trivial weights. In addition, the degeneracy problem can't be solved in the **SIS** framework. In order to cope with this degeneracy problem, a resampling step is suggested to be used after weight normalization. The following subsection presents the sampling importance resampling (**SIR**) method using resampling.

2.3.3 Sampling Importance Resampling (SIR)

Whenever a significant degeneracy happens, the **SIS** algorithm must use the resampling step. The idea of the resampling step is to eliminate the particles with insignificant weights and concentrate on particles with significant weights. It involves a mapping of the random measure $\{x_t^i, \pi_t^i\}$ into a random measure $\{x_t^{i^*}, \frac{1}{N}\}$ with uniform weights. The new set of random samples $\{x_t^{i^*}\}_{i=1}^N$, where N is the number of the samples, is resampled N times. One of the resampling algorithms is summarized in Table 2.2 (see [11] for more resampling algorithms). The important point to note here is that the resampling step does not change the distribution. The description of the resampling step is summarized in Table 2.2.

Table 2.2. Resampling Algorithm

$[x_t^{j^*}, \pi_{t\,j=1}^{j\,N}]$ = RESAMPLE $[x_t^i, \pi_{ti=1}^{i\,N}]$
- Initialize the CDF: $c_1 = \pi_t^1$
- FOR i = 2 : N
  - Construct CDF: $c_i = c_{i-1} + \pi_t^i$
- END FOR
- Start at the bottom of the CDF: i = 1
- Draw a starting point: $\mu_1 \sim Uniform[0, N^{-1}]$
- FOR j = 1 : N
  - Move along the CDF: $\mu_j = \mu_1 + N^{-1}(j-1)$
  - WHILE $\mu_j > c_i$
    i = i + 1
  - END WHILE
  - Assign sample: $x_t^{j^*} = x_t^i$
  - Assign weight: $\pi_t^j = N^{-1}$
  - Assign parent: $i^j = i$
- END FOR

There are many efficient resampling algorithms such as stratified sampling and residual sampling [11]. The most intuitive resampling method would be implemented by generating N i.i.d variables from the uniform distribution, sorting them in ascending order and comparing them with the cumulative sum of normalized weights. In this case, the complexity of the resampling algorithm is O(NlogN) when the best sorting algorithm is used.

The sequential importance sampling algorithm forms the basis for most particle filters that have been developed so far. The most popular versions of particle filters are as follows: sampling importance resampling filter, auxiliary sampling importance resampling, particle filters with improved sample diversity, local liberalization particle filters,

Table 2.3. SIR Particle Filter

$[x_t^i]$ = SIR $[x_{t-1}^i, z_t]$
- FOR i = 1 : N
  - Draw $x_t^i \sim q(x_t|x_{t-1}^i)$
  - Calculate $\pi_t^j = p(z_t|x_t^i)$
- END FOR
- Calculate total weight: $t = SUM[\pi_{t\,i=1}^{i\,N}]$
- FOR i = 1 : N
  - Normalize: $\pi_t^i = t^{-1}\pi_t^i$
- END FOR
- Resampled by the algorithm in Table 2.2 $[\{x_t^{j^*}, \pi_t^j\}_{j=1}^N]$ = RESAMPLE $[\{x_t^i, \pi_t^i\}_{i=1}^N]$

---

and multiple model particle filter. In our experiments, sampling importance resampling (**SIR**) was implemented and the pseudo code is summarized in Table 2.3.

CHAPTER 3

RELATED WORK

Target Tracking is required by many vision applications such as human-computer interfaces, video communication/compression or surveillance, and so on. Due to the observation uncertainty from sensors (cameras, laser ranger finders, infrared sensors, etc), tracking is a challenging task. In this context, particle filters (Bayesian filters) have proven very successful for non-linear and non-Gaussian estimation problems (i.e., they are neither limited to linear systems nor require the noise to be Gaussian). For such systems tied down with uncertainty and noise, the extensions of the particle filter to multiple target tracking have increasingly received attention. This thesis proposes an effective and efficient solution to multi-target tracking. In the contexts of multi-target tacking, the difficulty lies in the fact that the estimation of the states requires the assignment of the observations to the multiple targets. A number of research efforts and solutions to tracking multiple targets are extensions of the particle filters. However, most algorithms broadly fall into one of two categories. The first category consists of methods generating multiple instantiations of single object tracking, *e.g.* [10, 17, 9], whereas the second category of multi-target trackers explicitly extends the state space of each filter to include components for all the target distributions, *e.g.* [20, 18, 17, 16, 1]. This chapter briefly reviews the two different categories for multi-target particle filters.

3.1 Multiple Instantiations of Single Target Tracker

The approach to instantiating multiple single target trackers generally has been dedicated to the explicit interpretation of the resulting trackers in the case of occlusions

and overlapping targets. Jaco Vermaak, Arnaud Doucet, and Patrick Pérez [10] present multiple instantiations of single object tracker, called the *mixture* Particle filter. Each Particle filter is considered to track one target under an independent target assumption. This assumption is also adopted in this thesis. In [10], K-mean clustering is used to generate the potential hypotheses and each clustered area is assigned an individual filter. The clustering procedure is expensive in terms of computations. The new filtering distribution is a mixture of the individual component filtering distributions. Assigning the area of interest to the Particle filters plays an important role in the sense that the individual particle filters do not interact with each other. So the authors introduced the mixture weight which works as a data association scheme. However, this scheme fails when occlusion occurs and in those situations requires re-clustering after the targets are separated. The approach proposed here avoids the problem by explicitly modeling occlusion and by providing a mechanism for the creation and deletion of new target trackers.

As a choice of filtering method, [17] uses multiple Kalman Filters for tracking multiple objects. A number of restrictions on the video sequence that they made are that the sequence is captured using a stationary camera with a fixed focal length, if the sequence contains moving people, they are moving in a upright fashion, and the video camera is positioned such that the 3D point intersection between the optical axis and the ground plane projects onto the image plane. The final assumption provides a reasonable foundation for occlusion reasoning while using a stationary camera.

3.1.1 State Space Extension

The two general ways of extending the stat space are that the first dynamically changes the dimension of the state space and the second adds a set of indicator variables signifying whether an object is present or not. In [18], multiple target tracking consists of estimating the state vector created by concatenating the state vectors of all targets under

the following assumption: each target moves independently under Markovian dynamics. The key idea of this paper is the extension of the system state in the Particle filter. The state representation is the joint density: association vector (probability) and target identifier. Our proposed method does not extend its system state and since the projection between the particle space and the image space is computed as we evaluate a standard observation likelihood, the proposed method does not require any special dynamic model.

Blob trackers [1] have proven useful in tracking objects in a scene by modeling it using an elliptical contour and thereby segmenting it from the background. These algorithms have a severe performance bottleneck because background subtraction is usually followed by blob detection and tracking. [1] introduces two theoretical advances in enhancing the performance of multi-object trackers: one is a Bayesian correlation-based multi-blob likelihood function, and the other is a Bayesian Particle filter for tracking multiple objects when the number of objects present in a scene can vary. The authors use the CONDENSATION algorithm [7] for their purpose to demonstrate the utility of the Bayesian likelihood function. The CONDENSATION algorithm has been augmented here to track distinctly identified objects as they change positions over time. We use a different version of the Bayesian Particle filter, Sampling Importance Resampling (SIR) Particle filter. In [1], one problem faced by the tracker is when one object being tracked passes in front of the other; in such cases the labels assigned to the objects are switched. In such cases, the tracking algorithm fails to distinguish between different foreground objects. This shortcoming also happens at the proposed method because we only use the Color-based observation model.

The previous papers concentrate on single camera 2D tracking. [20] proposed probabilistic tracking in 3D and derived a mathematical observation formulation that is robust to occlusions. The state space is extended by concatenating its configuration into a single super state vector. With the 3D information in the state space, observations are

modeled with a joint observation model. However, since the joint particle filters suffer from the exponential computational cost in the number of targets to be tracked, the approach soon becomes impractical as the number of the targets increases. To circumvent this problem, they defined a probabilistic measure of similarity between the image and a synthesized view by associating to each image pixel a weight representing its reliability to belong to the object under analysis. This approach is similar to our proposed method except that we do not employ the 3D information into the state space and each normalized pixel value also implies the occlusion situations.

Christopher Rasmussen [23] presents a new *joint* target tracking algorithm, called the *joint* likelihood filter (JLF), which is based on the principles behind the Joint Probability Data Association Filter (JPDAF) but allows for tracked objects to overlap one another and deduces their depth ordering from the image whenever possible. The state space is extended using a framework that explicitly reasons about data association and the JLF approach allows mixed tracker when tracking several objects and accommodates overlaps robustly as well. The author also extended the JLF by combining geometric parts and qualitative modalities (color, shape, texture, etc) of tracked objects. The joint measurement method tends to lessen the ambiguity of the occlusion reasoning. A shape observation model has a strong geometric property to identify the occlusion situations. However, we do not use the shape likelihood filter but color-based observation model instead.

CHAPTER 4

DISTRIBUTED MULTI-TARGET PARTICLE FILTER

4.1 Introduction

This chapter describes the main contributions of this thesis. We show that the proposed method is capable of tracking multiple objects of the same and different types. Each filter works independently when targets do not occlude each other. It only adds a constant factor compared to a standard Particle filter in terms of computational complexity.

[10] describes the challenges and drawbacks of the standard particle filter such that the target with the higher likelihood value typically dominates the distributions, as a result disregarding the other modes that have the lower likelihood values. In the context of multi-target tracking, we must maintain each individual target effectively enough to handle the occluding situations in a complicated scene. In Chapter 3, we describe that the mixture Particle filter instantiates multiple single trackers and approximately estimates its state in a single flat distribution according to

$$p(x_t|Z^t) = \sum_{k=1}^{K} \omega_{k,t} p_k(x_t|Z^t) \tag{4.1}$$

where K is the number of targets and the mixture weights satisfy $\sum_{k=1}^{K} \omega_{k,t} = 1$ [10]. The *mixture* Particle filter is modeled as a K-component non-parametric mixture model. The mixture posterior distribution is equivalent to the summation of individual target distribution. However, since the summation of each target distribution makes the resulting distribution a multi-modal distribution, it is difficult to estimate its state. Under the

assumption that each target moves independently, the idea of a *mixture* Particle filter is ideally suited because each component (mode) is modeled with an individual Particle filter. However, the mixture distribution requires that each filter is modeling a different target and in the case of occlusion might lose track of one of the targets. To address this, we consider each independent filter per target to represent separate dimensions in a joint tracking distribution. The *mixture* Particle filters avoid the well known problem of "sample depletion", which is largely responsible for loss of track, by distributing the resampling step to individual filters [9]. We also adopt the multiple resampling step where each filter is resampled separately.

The computational cost in the *joint* particle filter increases exponentially in the tracked number of the targets due to the increase in the number of particles required to represent a higher dimensional distribution with equal precision. To circumvent this problem, if we assume that the targets never occlude each other, we could simply run one individual particle filter for each target, thus leading to no overhead in the computational cost. However, assumptions that tracked targets always evolve independently and never occlude each other are not practicable options. The proposed particle filter includes an observation model that explicitly represents partial and complete occlusion situations by factorizing $p(z_t|x_t)$. The factorized observation models efficiently and effectively represent the probability function for an object being visible at the known location and for an object not being visible given that the target is hidden. The method will be explained in detail in Section 4.2.2.

Consequently, the main contributions fall into three parts in this thesis. In the first contribution of this thesis, in contrast to the Bayesian filtering distribution of the *mixture* Particle filer in [10] which is derived by the summation of each filter, the basis of

our filtering distribution is the joint Bayesian distribution under a limited independent assumption. Ideas similar to the *mixture* particle filter are used to overcome the drawback of Monte Carlo methods that are poor at consistently maintaining the multi-modality of the target distributions that may arise due to ambiguity or the presence of multiple targets. However, we extend this method by distributing the filters to estimate the joint Particle filter with a linear computational cost.

As the second contribution, a new joint observation model is proposed by factoring the standard observation model into two parts: the probability of the target being visible plus the probability of the target not being visible. To maintain multiple regions of interest even under occlusion situations, the proposed approach includes occlusion reasoning. This approach not only explicitly represents an occlusion situation, but also performs approximately as well as exponentially complex joint filter under the following assumptions: (1) targets move independently, (2) targets are rigid, and (3) each pixel in the image comes from only one of the targets.

As the third contribution, target creation is proposed by running an additional filter, called "background filter". The background filter plays a role in surveilling a new target and in the case that the new target is found, a new filter is created without increasing complexity and biasing other target distributions. Target deletion is also proposed to destroy a filter that is tracking a target not being visible in the current scene. However, when the target is occluded by other targets, then the tracked filter is not destroyed. The threshold is defined for the termination of a tracking filter.

4.2 Multi-Target Filtering Distributions

This section describes how the filtering distribution is modeled to express multiple independent filters and how to approximate the joint observation likelihood model, which

is often intractable in a *joint* Particle filter, with multiple instantiations of a single target tracker.

4.2.1 Target Filtering Distribution

The standard particle filter approximates the posterior distribution $p(x_t|Z^t)$, where x_t is the current target state at time step t, given all observations $Z^t = (z_1, ..., z_t)$ up to the current time step. How can we model multiple targets? Before we explain the proposed method, we show the multi-target filtering distribution as a form of Bayesian filter $p(x_t|Z^t)$ and some notations.

Due to the sample depletion problem in the standard Particle filter as shown in Chapter 5, it is impossible to track multiple targets with the same single distribution. In the context of tracking multiple targets, we can circumvent this problem by extending the posterior distribution $p(x_t|Z^t)$ that is recursively updated over the joint state of all **K** targets, i.e. $x_t = (x_{1,t}, ..., x_{K,t})$ given all observation $Z^t = (z_1, ..., z_t)$ up to and including t under the assumptions that the number of targets to be tracked, **K**, is fixed and known, the measurements are independent, and the individual targets evolve independently according to the dynamic model of each target $p_k(x_{k,t}|x_{k,t-1})$, k = 1 ... K. Thus, the Bayesian multi-target distribution is defined as follows:

$$
\begin{aligned}
p(x_t|Z^t) &= \prod_{k=1}^{K} p_k(x_{k,t}|Z^t) \\
&= \frac{\prod_{k=1}^{K} p_k(z_t|x_{k,t}) p_k(x_{k,t}|Z^{t-1})}{\prod_{k=1}^{K} \int p_k(z_t|x_{k,t}) p_k(x_{k,t}|Z^{t-1})} \\
&= \frac{\prod_{k=1}^{K} p_k(z_t|x_{k,t}) \int_{x_{k,t-1}} p_k(x_{k,t}|x_{k,t-1}) p_k(x_{k,t-1}|Z^{t-1}) dx_{k,t-1}}{\prod_{k=1}^{K} \int p_k(z_t|x_{k,t}) \int_{x_{k,t-1}} p_k(x_{k,t}|x_{k,t-1}) p_k(x_{k,t-1}|Z^{t-1}) dx_{k,t-1}}
\end{aligned}
\tag{4.2}
$$

The observation likelihood function, $p_k(z_t|x_{k,t})$, evaluates the measurement probability z_t given a particular state of $x_{k,t}$ at time t. The observation models in each target state are also independent of each other. The dynamic model of each target, $p_k(x_{k,t}|x_{k,t-1})$, predicts the state $x_{k,t}$ at time t given the previous state $x_{k,t-1}$. The above filtering distribution overcomes the curse of dimensionality by recursively updating the marginal filtering distributions of $p(X^t|Z^t)$through the Bayesian sequential estimation recursion under the independence assumption mentioned above.

The last point to note here is how to deal with the resampling procedure in the above context. The standard Particle filter resamples the particles to avoid the degeneracy problem as mentioned in Chapter 2. Since each Particle filter can linearly compute the resampling step, we treat each target distribution independently such that it allows independent resampling of each of the Particle filters.

4.2.2 Joint Observation Likelihood Model

In general, to deal with multiple targets in the *joint* pdf $p(x_{k,t}|Z^t)$, where k = {1 ... K}, is a correct but intractable option. In Section 4.2.1, we developed the *joint* filtering distribution, which forms the filtering basis of this thesis under an assumption that each filter works independently. In this section, we show the factorization of the standard

observation model and the explicit interpretation of each factorized term to relax the independence assumption to allow for target occlusion.

As shown in the Bayesian sequential estimation framework in Chapter 2, the filtering distribution can be computed according to the two step recursion: prediction and update step. If we, in addition to assuming that each target moves independently, also assume that targets never occlude each other, then the proposed filter will fail when the targets pass over each other or are occluded for a while. Under the independent filter assumption, each particle filter samples in its own target distribution and the complexity will be linear in the number of targets. However, it would be a bad assumption that targets never occlude each other. What follows is a derivation of the prediction and update step used in the proposed approach.

The prediction step can proceed independently for each target as long as no collision situations occur as follow:

$$p_k(x_{k,t}|Z^{t-1}) = \int_{x_{k,t-1}} p_k(x_{k,t}|x_{k,t-1}) p_k(x_{k,t-1}|Z^{t-1}) dx_{k,t-1} \tag{4.3}$$

However, the update step is not suited to represent occlusion situations. Due to the filter association to measurement (since we assume that each filter represents one target), the update step can not be performed independently for each target. To overcome this challenge, we factor the observation likelihood model of each target $p_k(z_t|x_{k,t})$ according to

$$p_k(z_t|x_{k,t}, F_{1\ldots K}) = p_k(z_t|x_{k,t}, V_k) p_k(V_k|x_{k,t}, F_{1\ldots K}) + p_k(z_t|x_{k,t}, \overline{V_k}) p_k(\overline{V_k}|x_{k,t}, F_{1\ldots K}) \tag{4.4}$$

where $F_{1...K}$ denotes the **K** target filters and V_k indicates target k being visible and $\overline{V_k}$ indicates target k not being visible in location $x_{k,t}$. Equation 4.5 consists of four different terms where each term functions differently: (1) the observation likelihood $p_k(z_t|x_{k,t}, V_k)$, (2) likelihood of target k being visible, $p_k(V_k|x_{k,t}, F_{1...K})$, (3) the expected value of the observation model $p_k(z_t|x_{k,t}, \overline{V_k})$, and (4) the last term is equivalent to 1 - $p_k(V_k|x_{k,t}, F_{1...K})$. In other words, the observation model is defined in terms of a probabilistic measure of the particle weights and normalized particle weights originating from all the pixels belonging to the samples of the filter. In [20], Oswald Lanz used a visibility method to define $p(z|x)$ as a probabilistic measure of similarity between the noisy image and the synthesized view. Similar to our observation model, the author derived this observation model based on image pixels with a target state encoding 3D information. If a filtering distribution represents a state of a target correctly, the state of each target can indicate the possible occlusion because of the 3D information and the likelihood of all other targets that are hidden does not need to be computed. Since we use a similar approach using 2D information and the ordering of the particles cannot be computed, we include the occlusion likelihood in the second term by normalizing the pixel likelihood on each pixel location with a summation of all the contributions of all the particles.

What follows is a detailed interpretation of the second and the third term. Before we explain these terms, the followings are the interpretation of the first and the fourth term. The first term $p_k(z_t|x_{k,t}, V_k)$ computes the measurement likelihood of each particle given that target k is visible at the given location, which is similar to the standard observation likelihood model. The fourth term is easily obtained from the second term as follows:

$$p_k(\overline{V_k}|x_{k,t}, F_{1...K}) = 1 - p_k(V_k|x_{k,t}, F_{1...K}) \tag{4.5}$$

4.2.2.1 Visibility Likelihood: $p_k(V_k|x_{k,t}, F_{1...K})$

In the probabilistic exclusion principle [18], a state was extended to contain two objects: a foreground object and a background object. In a boosted particle filter [17], the cascaded Adaboost algorithm was used to provide a mechanism for allowing objects leaving and entering the scene effectively. Similarly, [10] used the K-means spatial reclustering algorithm for maintaining the targets by merging and splitting the clusters. Since we do not use any clustering methods or do not extend the target states, which may cause the complexity cost to grow exponentially, we need to provide an efficient and effective method to obtain and maintain the target representation. To do this, we propose a new method to evaluate the visibility likelihood in Equation 4.6. This method consists of three steps: (1) project particles into the image space by accumulating each particle weight over the corresponding pixels, (2) normalize each pixel with the summation of the accumulated weights from each target filter, (3) project back from the image space to the particle space by re-normalizing the pixel weights over the target area of the corresponding particles. What follows is a mathematical derivation of the three steps.

The visibility likelihood term is derived by using Bayes' law as follows:

$$p_k(V_k|x_{k,t}, z_t) = \frac{p_k(z_t|x_{k,t}, V_k)p_k(V_k|x_{k,t})}{p_k(z_k|x_{k,t})} \tag{4.6}$$

The main idea of the first step in Equation 4.6 is to project the particle weights into the image space. In other words, we translate this term to the following equation:

$$\alpha p_k(V^k_{PixWeight}|X^k_{PixLoc}, z_t) = \sum_{x_{k,t}|X_{PixLoc} \in Pixel(x_{k,t})} p_k(z_t|x_{k,t}, V_k) \tag{4.7}$$

where $Pixel(x_{k,t}) = \{X^K_{PixLoc}\}$, which is an area of interest originated from each particle in image space. Equation 4.7 is nothing but the pixel weight on all pixel locations belonging to target k. We assume that each pixel can only be from one target and each target equally likely appears in the foreground. In practice, if the size of the filter is m by n, then we project each particle weight to the m by n area of pixels as illustrated in Figure 4.1. In other words, the particle weights are accumulated on each pixel and eventually if the particles are close to each other, the region will have the highest particle weight.

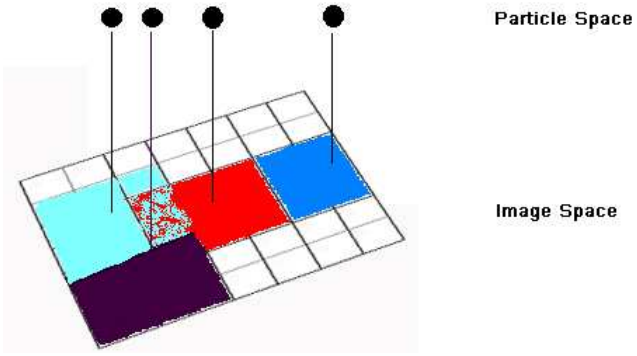


Figure 4.1. Given the particle weight, each particle weight is projected into the image space evenly according to the size of the filter.

After projecting each particle weight to the corresponding pixels, since we deal with multiple targets, the image plane for each target filter, which contains of the accumulated particle weights, is separately generated for normalization purposes. With the multiple image planes, we take care of the interpretation of the occlusion situations by normalizing each pixel with the summation of all the pixel weights from all the filters. The normalizing factor α can be determined by deserving that the following condition has to hold as follows:

$$\sum_{k=1}^{K} p_k(V^k_{PixWeight}|X^k_{PixLoc}, z_t) = 1 \tag{4.8}$$

Each pixel weight after normalization is indicating how likely the pixels under analysis can come from any of the **K** targets. In this context, the scalability assumption should hold because if the number of particles for the first target is 50 and for the second target is 100, then the importance of the region for the second target is twice as strong as that of the first target. Consequently, the normalization factor (Equation 4.8) gives the proportional importance weight to each pixel across each target according to the following equation:

$$p_k(V^k_{PixWeight}|X^k_{PixLoc}, z_t) = \frac{1}{\alpha} \sum_{x_{k,t}|X_{PixLoc} \in Pixel(x_{k,t})} p_k(z_t|x_{k,t}, V_k) \tag{4.9}$$

where $\frac{1}{\alpha}$ is a normalization factor. All the pixels should get normalized with the summation of each pixel corresponding to each image plane,

$$\alpha = \sum_{k=1}^{K} \left(\sum_{x_{k,t}|X_{PixLoc} \in Pixel(x_{k,t})} p_k(z_t|x_{k,t}, V_k \right) \tag{4.10}$$

After each pixel gets normalized to satisfy Equation 4.8, we need to project the image space back to the particle space. Therefore, we need another normalization factor, corresponding to the image area associated with the particles. The following equation shows how the normalized pixel values over all the image planes get assigned into the second term.

$$p_k(V_k|x_{k,t}, z_t) = E_{X_{PixLoc} \in Pixel(x_{k,t})} \left[p_k(V^k_{PixWeight}|X^k_{PixLoc}, z_t) \right] \tag{4.11}$$

In particular, in the experiments performed here the pixel area of a particle in represented by the 11 by 11 pixel area of the color template.

4.2.2.2 Observation Model For Hidden Targets: $p_k(z_t|x_{k,t}, \overline{V_k})$

The third term of Equation 4.4 requires a method for evaluating the measurement support (likelihood) of occluded targets, whereas the first term is relatively easy to evaluate the measurement support of each target being visible. The important point of this term is how to maintain the filtering distribution of each target without biasing the resulting distribution and how to exploit the observation information even if the targets are occluded by other targets or clutter. For this, we compute the third term not only not to assign too much weight to occluded target,but also not to move too much weight to the foreground target from the targets behind, by utilizing the expectation of the measurement model $p_k(z_k|x_{k,t}, \overline{V_k}) = E_{z_t}[p_k(z_t|x_{k,t}, V_k)]$.

4.2.3 Filter Deletion

This section presents the fundamental framework of filter deletion, which happens when a target disappears out of the scene. Filter deletion does not automatically happen in the cases where one target is located behind another target for a long time because the invisibility likelihood explains an occlusion situation. But if the target being tracked suddenly disappears out of the scene, then we initiate the filter deletion. As a result, the filter used to track the targets should be destroyed for the sake of process efficiency. Since the filters are not smart enough to know when to terminate themselves, we utilize information to compute the proposed observation likelihood. In other words, we determine the likelihood , $p_k(F_k^t|Z^t)$, that the target associated with filter k exists and define a threshold to decide when to destroy the target distribution upon the disappearance out of the scene. The following equation is the concept of the likelihood that a target exists.

$$p(F_k^t|Z^t) = \frac{p(z_t|F_k^t, Z^{t-1})p(F_k|Z^{t-1})}{p(z_t|Z^{t-1})} \tag{4.12}$$

where the numerator indicates particle weights $\sum_k^i \pi_i$ and the denominator is the expected value of observation model in the current filter $E[p(z_t|F_t)]$. This value obtained from Equation 4.12 together with a threshold is used to determine whether the tracked target filter is destroyed or not.

4.2.4 Filter Creation

Creation of another target filter is achieved through the addition of an additional filter, called background filter. The background filer has to track what other filters do not track. In order for this background filter not to represent the target of the other filters, we use the visibility likelihood model described in section 4.2.2.1, but do not apply the normalization procedures so that the background filter never bias on visibility likelihood of other targets.

Each particle in the background filer randomly gets assigned one of the measurement models being tracked. In order to prevent the background filter from tracking targets already traced by other filters have to be considered, operation happens in the image plane by removing all the pixels in the image plane of the background filter which cover regions that are already being tracked by other filters. This means that we never find objects that are partially overlapped by others. Then, how do we find a cluster that might be a potential target that other filters are not tracking? The simplest solution for detecting the cluster that other filters are not tracking is to use a density estimate by computing the highest pixel weight of the background filter in the image space and

compare it to a stationary threshold. If the weight of the cluster in the image plane of the background filter does exceed the significance threshold, we create a filter for that target. The important element to note here after finding new target is that we have to transform the current state of the background filter to the new filter including the observation model.

To make target finding more efficient and to avoid the depletion problem, once the background filter falls into one of the targets that has already been tracked by other filters, we bias the distribution of the background filter by resampling a half or 60 percentage of the particles in the resampling step and replacing the rest with random new particles so that it easily starts to search a new target.

4.3 Computational Complexity

With the assumption that a target is a rigid object and not transparent, the key to a robust multi-target occlusion observation model in the proposed method is to project the particle weight to each corresponding pixel and introduce a normalization factor to include occlusion situations. As particles in the joint particle filter contain the joint position of all targets and the filter thus suffers from exponential complexity with the increasing number of targets, we proposed a pixel-based computation technique not only to help the proposed observation likelihood model (Equation 4.2) to explain the possible occlusions, but also to reduce the complexity of the filter to linear in the number of targets.

The important point to note in the proposed method is that the computational complexity does not increase exponentially even though the number of targets is increased. In this thesis, the only element that is required to be computed is the interaction of the importance weights. This means that we only need to worry about the observation

model, which is the first and second term in Equation 4.5. How do we compute these terms without increasing the computational complexity? In what follows, we first discuss why the *joint* particle filter is intractable and inefficient when it handles the occlusion situations and discuss how the proposed method with its observation model provides a computation cost that grows linearly while approximating a joint distribution. by utilizing the pixel-based computation.

The typical *joint* particle filter suffers from the curse of dimensionality because each particle is Kd dimensional, where d is the dimensionality of an individual filter and estimates the components of all the targets' state being tracked. Consequently, if the particle filter needs N particles, then N^K are typically required for tracking K targets with the same precision. In addition, if we define M as the size of the the template/object area, then the resulted complexity of the *joint* particle filter becomes $\boldsymbol{O}(MN^K)$ under the assumption that resampling step is $\boldsymbol{O}(N)$, where N is the number of particles. Thus, tracking multiple targets increases the computational cost and space exponentially. To get around this difficulty, Section 4.2.1 hypothesizes that under the stated assumption the proposed multi-target filter is suited to represent the target distribution. If the resampling step is again assume that we linear compexity, then the total complexity becomes $\boldsymbol{O}$(4MNK) because the standard particle filters takes $\boldsymbol{O}(MN)$ and the proposed observation model takes $\boldsymbol{O}(3MNK)$: $\boldsymbol{O}$(MNK) for projecting the particle space into the image space, $\boldsymbol{O}$(MNK) for normalization and $\boldsymbol{O}$(MNK) for projecting back to the particle space. Figure 4.2 illustrates the projection between the particle space and the image space.

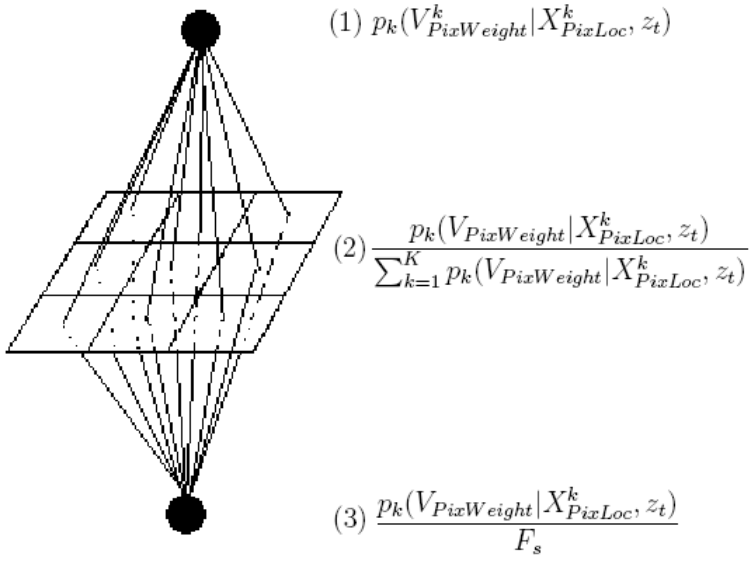


Figure 4.2. Projection step between the particle space and the image space. (1) Projection into image plane. (2) Normalization of each pixel. (3) Re-normalization of all the normalized pixels..

4.4 SIR Multi-Target Particle Filter

This section describes the detailed steps of the distributed multi-target Particle filter algorithm.

- INITIALIZATION

  At time t-1, the state of the targets is represented by a set of unweighted samples $\{x^i_{k,t-1}\}^N_{i=1}$, where N is the number of the samples. Each target has a set of these samples.

- FOR EACH TARGET k : K

  1. Evaluate the observation likelihood $\{x^i_{k,t}\}^N_{i=1}$.

2. Construct the Image Plane for each target k by normalizing each pixel in each image plane according to Equation 4.9 and Equation 4.11.
3. Calculate the expected value of the measurement model.
4. Calculate the fourth term in Equation 4.5 : 1 - $p_k(V_k|x_{k,t}, F_{\forall_K})$.
5. Compute the importance weight distribution $\{\pi_k^i\}_{i=1}^N$ by assigning the values from the above four steps into Equation 4.5.

- END

- RESAMPLING STEP
  - FOR EACH TARGET k : K
    - Resample each target state $\{x_{k,t}^i\}_{i=1}^N$ according to Table (2.2).
  - END
- END
- The resulting sample sets $\{x_{k,t}^i, \pi_{k,t}^i\}$ for each target at time t represents an estimated state of the targets.

CHAPTER 5

EXPERIMENTAL RESULTS

This chapter describes the state-space models, the observation models, and expectation of the observation model. For experiments, we compare the performance of the proposed multi-target particle filter with that of the standard particle filter on three different tracking problems. The first experiment demonstrates the standard Particle filter over multiple targets. The second are three tracking examples with different numbers of targets using the proposed multi-target Particle filter. The last shows examples of filter deletion and creation. All the experiments are performed on a video sequence.

5.1 State Space Model

There is significant literature related to multi-target tracking. Even though all the literature has different ways to address the tracking problem, the way of approximating the state space generally falls into one of two categories. The first uses the general way such that the states consist of the image coordinates (*i.e.* xy plane), velocity, or acceleration of targets, *e.g.* [2,3]. In the context of multi-target tracking we assume that the number of targets to be tracked, $\mathbf{K}$, varies and is known *a priori*. Each target is parameterized by the state $x_{k,t}, \{k = 1...\mathbf{K}\}$, which has different configurations (i.e. different image coordinates) for the individual targets. The representation of multiple targets X_k is given by the individual target states, i.e. $X_k = (x_{1,t}, \ldots, x_{K,t})$. In the experiment, the targets are rectangular boxes. We therefore define the particle representation of each state as $x_{k,t} = \{x_t, y_t\}$ where x, y specify the location of the samples. The

particle representation has two degrees of freedom, one in each direction of the image coordinates.

5.2 Observation Likelihood Model

This section describes the likelihood model $p_k(z_t|X_{k,t})$ where z_t is a likelihood evaluated by the observation model (color-based observation model in the experiment) and $X_{k,t}$ is the position of objects. In short, the observation likelihood model $p_k(z_t|X_{k,t})$ expresses the likelihood of target k given that the objects would be located at $X_{k,t}$. The observation likelihood values (*i.e.* probabilistic measurement of similarity between $z_{k,t}$ and reference features) do not only arise from the targets to be tracked, but also additional clutter likelihood may result due to spurious objects, background changes, etc. We will assume that each of the targets can generate at most one likelihood at a particular time step. The likelihood model we will use for our experiments is a color likelihood model that was proposed in [15, 24] and the following section explains it in detail.

5.2.1 Color Likelihood Model

The color likelihood model [15] generates a measurement of similarity by comparing the color histogram of candidate regions in the current scene to a reference color histogram. We build color models by utilizing the color histogram method in the Hue-Saturation-Value (HSV) color space in order to decouple chromatic information from shading effects. We measure the likelihood through the Bhattacharyya distance between the two HSV color histograms of the reference and the candidate models.

Suppose that the distributions of the color histogram are transformed into m bins. The function $h(x_i)$ produces the histograms that associate the color at location x_i to the corresponding bin. In our experiments, we transform the histogram in RGB color space

to a HSV color histogram using $8 \times 8 \times 4$ bins to make the histogram less sensitive to lighting conditions.

The HSV color distribution of a reference model $\alpha = (\alpha_1, ..., \alpha_B)$ where B is the number of bins. It is defined as follows:

$$\alpha_b = \frac{1}{N} \sum_{i=1}^{N} \delta[h_{x_i} - b] \tag{5.1}$$

where N is the number of pixels of the reference region, δ is the Kronecker delta function, and $\frac{1}{N}$ is a normalizing factor so that $\sum_{b=1}^{B} \alpha_b = 1$.

Similarly, to construct the target model β_b, the RGB pixel values are retrieved from a region of interest using the state vector $\{x_{k,t}\}$ and transferred into HSV pixel values. Then the color distribution β_b of the candidate color model at time t is constructed similar to Equation 5.1 as follows:

$$\beta_b = \frac{1}{N} \sum_{i=1}^{N} \delta[h_{x_i} - b] \tag{5.2}$$

Given the distributions of two color models, the color model β_b associated with a hypothesized state $\{x_{k,t}\}$ will be compared to the reference model α_b. In our experiments, to measure how similar the candidate model is to the reference model, a similarity measure D is derived based on the Bhattacharyya similarity coefficient [15] and defined as

$$D[\alpha, \beta^t] = \sqrt{1 - \sum_{i=1}^{B} \sqrt{\alpha_i \beta_{t,i}}} \tag{5.3}$$

where B is the number of bins. As a result, the smaller D is, the more similar the distributions are. The similarity measure D is called the Bhattacharyya distance.

Since the smaller distance corresponds to the larger weight, we use the normal distribution to evaluate the likelihood between the two distributions. In the context of the particle filter, the weight π_k^i in the samples (x_k^i, π_k^i) is evaluated using the following normal distribution:

$$\pi_k^i(D) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{D^2}{2\sigma^2}} \tag{5.4}$$

where the width of the likelihood is controlled by the variance parameter σ^2 in the function of D. In our experiments, this standard deviation is assigned $\frac{1}{7}$. A similar model was already implemented in the context of the object tracking in [15, 21, 25]. Note that as the Bhattacharyya distance D can only take values between 0 and 1, $\pi_k^i(D)$ does not strictly represent a probability density, but would have to be scaled using $\frac{1}{\int_0^1 \pi_k^i(D)}$. Due to the normalization in the filter, however, constant is not required during calculation.

5.2.2 Compute Expected Observation Model

Chapter 4 explains the filtering distribution in a form of a probability density function (pdf) $p(x_{k,t}|Z^t)$ where k = {1 ... K} is the number of the targets. In particular, Equation 4.5 requires computation of the four related terms. This section shows a mathematical solution of the third term $(p_k(z_t|x_{k,t}, \overline{V_k}))$, which is an expectation of the observation model $E_{z_t}[p_k(z_t|x_{k,t}, V_k)]$ described in Chapter 4. We mathematically evaluate the expected value of the observation model as follow:

$$
\begin{aligned}
p_k(z_t|x_{k,t}, \overline{V_k}) &= E_D\left[\pi_k^i(D)\right] \\
&= \int_0^1 \pi_k^i(D) p(D)\, d_D \\
&= \int_0^1 \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{D^2}{2\sigma^2}}\right]^2 d_D \\
&= \frac{1}{2\pi\sigma^2} - \sigma^2 \left[e^{-\frac{(D^2)}{\sigma^2}}\right]_0^1 \\
&= \frac{-1}{2\pi}\left(e^{\frac{-1}{\sigma^2}} - 1\right)
\end{aligned}
\tag{5.5}
$$

where σ is given $\frac{1}{7}$, the range of a function of D is [0,1], and $p(x)$ is considered as the same distribution of $\pi_k^i(D)$ in this case. However, the expected value obtained from Equation 5.5 might not be a right value because our assumption is that Equation 5.4 is the distribution in a real image and therefore integrates to one because it is a probability function. This, however, is not the case here because we are using only a part of the Gaussian distribution. Eventually, we scale $p(D)$, the second term of the second line in Equation 5.5, to be a density function. However, since $p(D)$ does not measure the exact distribution in a real image, we alternatively determined this value by experiments. Eventually, this value is defined as 0.0000036 in the standard Particle filter and used for the expected value of the observation model. For the practical experiments, we use slightly lower value than 0.0000036 to bias the filter toward targets and then to reduce the time required for the particles to collapse into each target.

5.3 Experimental Results

As a benchmark, we compare the proposed multi-target Particle filters against standard Particle filters using the same objects. In the standard Particle filters, the experiments are performed by tracking multiple targets and the results are the precision in

terms of tracking accuracy. To compute the precision of tracking the targets, we manually track the center positions of each target and compare them against the expected value of the locations of the samples. The density estimate is used to calculate the position of the targets. Since it is well known that the precision of tracking gets much better with the increase in the number of the particles [22, 8], we do not include experimental results with different number of the particles. Instead, we focus on the precision and capability to track multiple targets as the number of targets increases. Targets are rectangular boxes in different colors (i.e., red, green, and blue) on image sequences recorded at a constant frame rate of 15 frames per second and with images of size 320 by 240 in RGB color space. The template used for the observation model consists of 11 by 11 pixels. The experiments for the proposed algorithm are performed with different numbers of targets. We mainly evaluate the precision in terms of the number of targets and the tractability of the occlusion situations. Finally, we show experimental results for a filter deletion and creation. Note that some experiments with high particle numbers did not run in real time.

5.3.1 Standard Particle Filters

In this experiment, the standard Particle filter runs to track the rectangular boxes of the same color. The targets move around in the cluttered background and sometimes disappear and appear again as shown in Figure 5.1. Below is the specific implementation information.

- The state $x_{k,t}$ of the k^{th} target consists of its position $(x_{k,t}, y_{k,t})$ in the image.
- For the observation likelihood model, we used an appearance template approach. In particular, we used a 11 by 11 square template containing the reference image transformed into HSV color space.

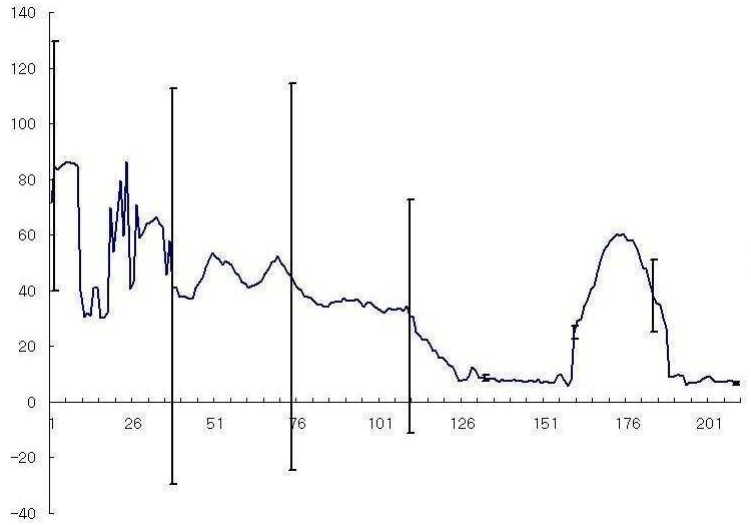
- For the dynamic model, we used a *Brownian* motion model, with mean 0.5 and standard deviation 0.2 for the proposal distribution in Equation 4.3.
- All filter used 300 particles.

Figure 5.1 shows the average errors in Cartesian distance between the estimated state and the true state and their standard deviations as error bars for tracking two targets after multiple runs using the standard Particle filter. The standard Particle filter does not maintain its estimated state consistently and has a high standard deviation at the beginning on both figures because the standard Particle filter sticks to one target at first and jumps back to the other target.

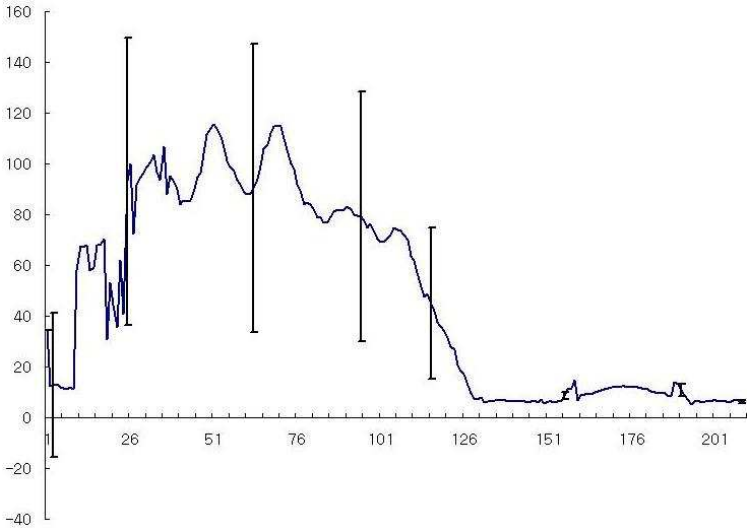
Figure 5.2 shows the sample depletion problem as described in Chapter 4. In the first frame, the particles spread out almost evenly to the two targets and the corresponding histogram of the X coordinate shows the two peaks indicating the particles stay at the two targets. However, as the filter runs, the number of particle on the left target keeps decreasing and at frame 30 and thereafter all particles migrate to the left target. This phenomenon demonstrates that the standard Particle filters are not suited to multi-target tracking.

Figure 5.4 and Figure 5.5 are the sequences of images showing the target disappearance using the standard Particle filter. Due to the sample depletion problem, we can't demonstrate the target occlusion situation. Therefore, this experiment only shows that the particles are severely diffused when the filter does not get any measurement support. Similar to the target disappearance phenomenon, when the target is occluded, the filter does not get enough measurement support to hold the importance weight distribution, thus resulting in spreading out of the particles. These experiments are also demonstrated

in the next subsection over distributed multi-target Particle filters. At frame 37 of Figure 5.2, when the target is about to disappear, the particles start to spread. At frame 51 of Figure 5.3, the diffusion of the particles is very severe and finally they are back on the target when the target appears again at frame 65.



(a) Target 1



(b) Target 2

Figure 5.1. Mean of errors in Cartesian distance and error bars using standard Particle filter on target 1 (a) and target 2 (b).

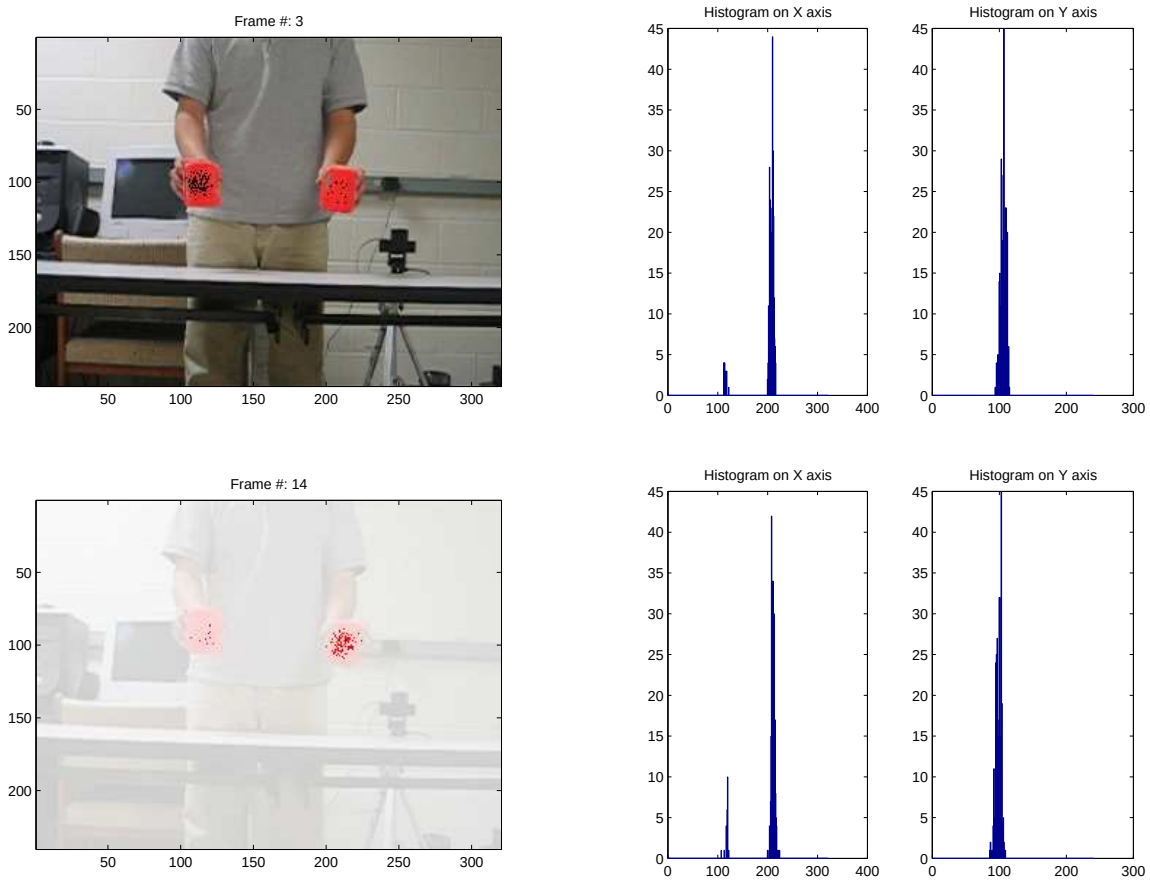


Figure 5.2. Sequence of images, demonstrating the sample depletion problem using standard Particle filter with 1000 particles.

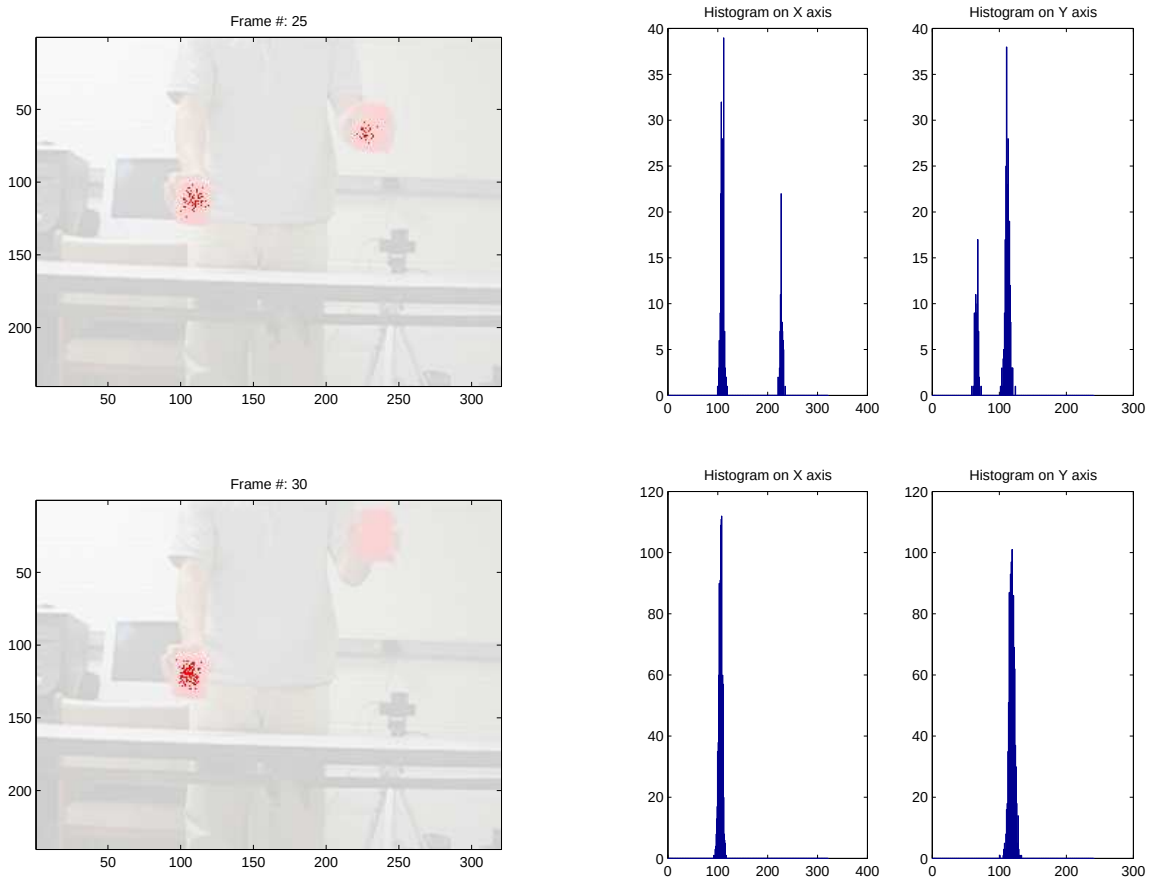


Figure 5.3. Sequence of images, demonstrating the sample depletion problem using standard Particle filter with 1000 particles.

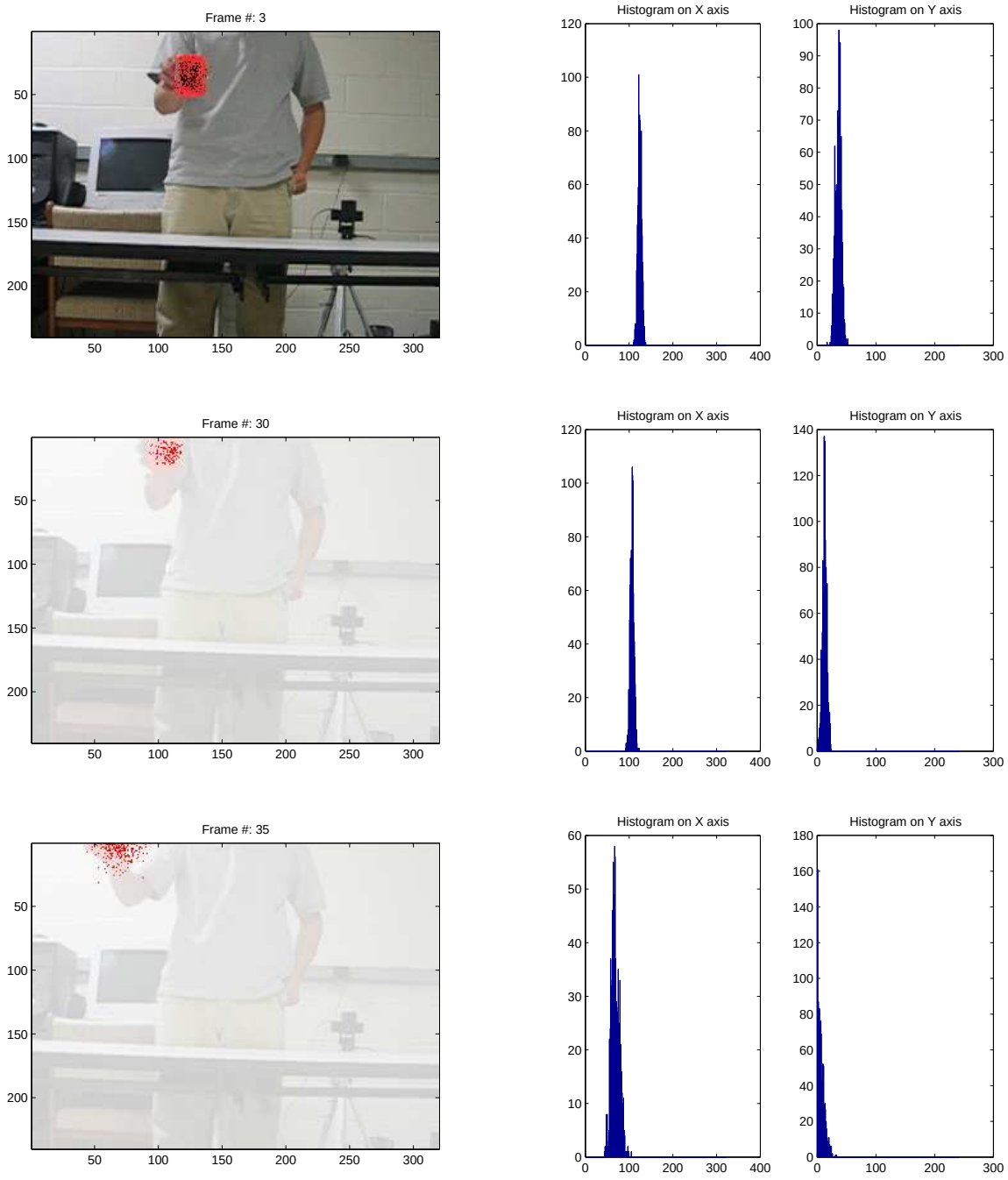


Figure 5.4. Target disappearance experiment using standard Particle filter with 1000 particles.

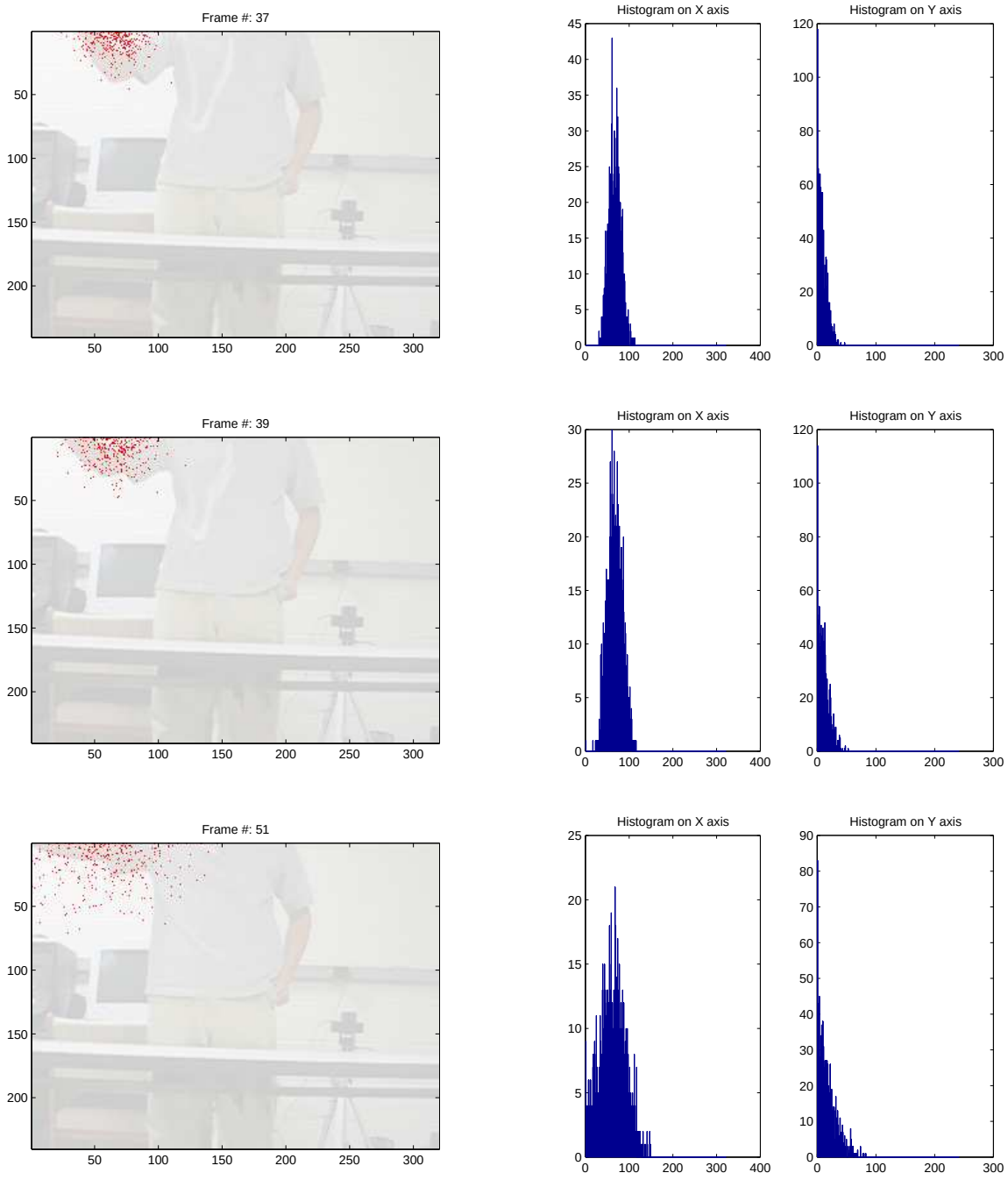


Figure 5.5. Target disappearance experiment using standard Particle filter with 1000 particles.

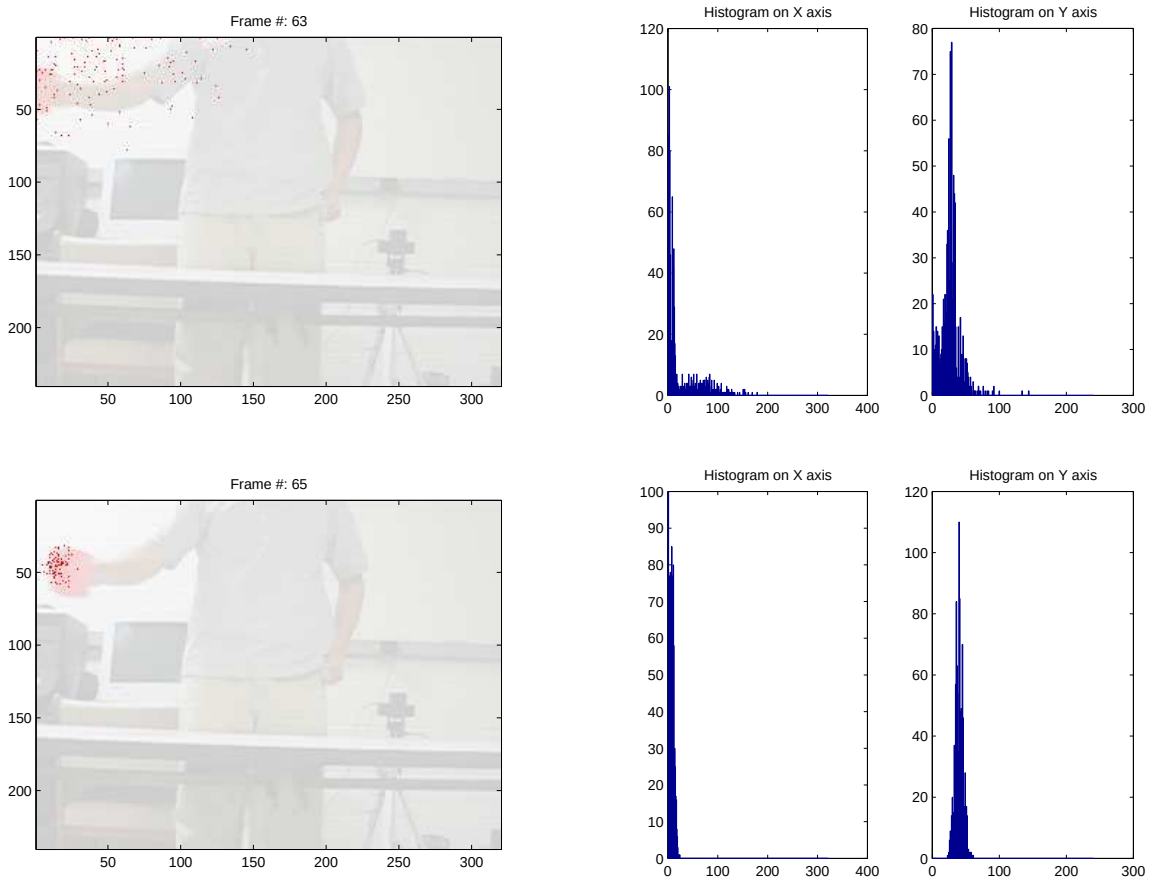


Figure 5.6. Target disappearance experiment using standard Particle filter with 1000 particles.

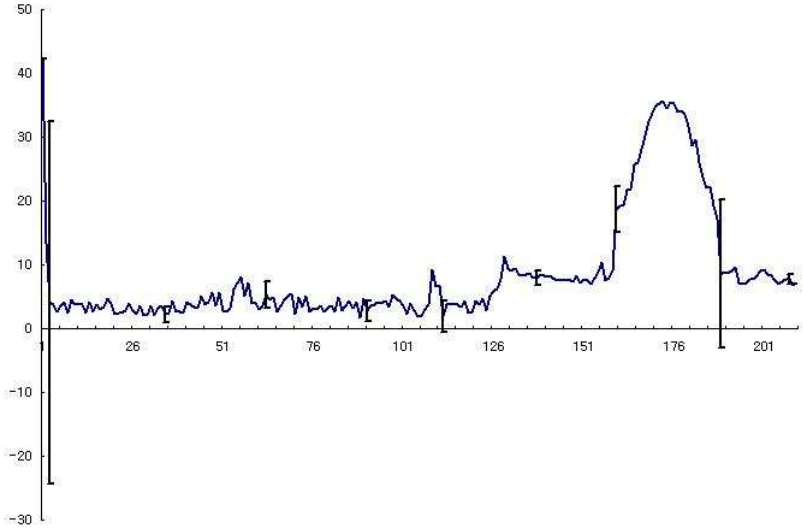
5.3.2 Distributed Multi-Target Particle Filter

The proposed algorithm is implemented according to the pseudo code algorithm in Section 4.4. The experimental setup is the same as that the standard Particle filter. This subsection consists of three different experiments: tracking two targets, tracking three targets, and target deletion and creation. Figures mainly illustrate the precision of the target trajectories against the true target trajectories and occlusion situations. The true state is manually estimated.

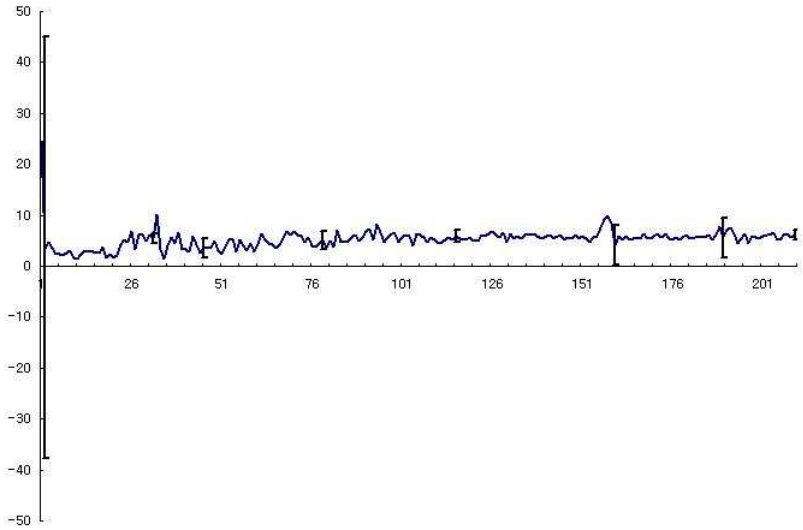
5.3.2.1 Tracking Two Targets of Different Color

Figure 5.7 shows the same information as Figure 5.1 but using the proposed multi-target Particle filter. The proposed distributed multi-target Particle filter is capable of tracking all the targets by maintaining low standard deviation between a true state and an estimated state. Therefore, it is well suited to tracking multiple targets.

Figure 5.8 and Figure 5.9 describe trajectories of a red and green target in the X and Y axis. Figure 5.10 to Figure 5.13 show a sequence of tracking images and their corresponding visibility and invisibility density for each target. While the green target passes behind the red target, the invisibility probability density explains the occlusion situation by increasing the density when the occlusion happens, and otherwise decreasing the density in case of no occlusion. Figure 5.10 (b) indicates that Invisibility density starts to increase and after the green target is not occluded its density dramatically drops to almost zero.



(a) Target 1



(b) Target 2

Figure 5.7. Mean errors in Cartesian distance and error bars using multi-target Particle filter on target 1 (a) and target 2 (b).

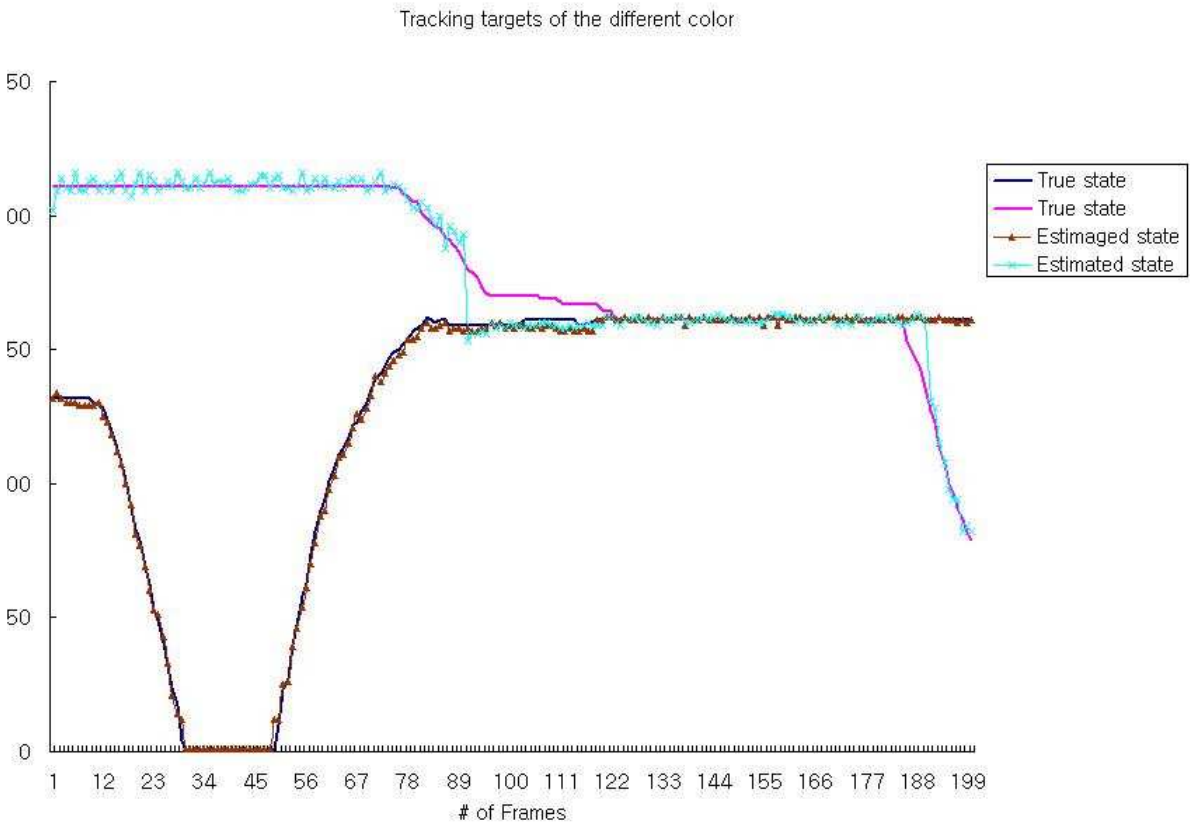


Figure 5.8. Tracking two targets int the X axis.

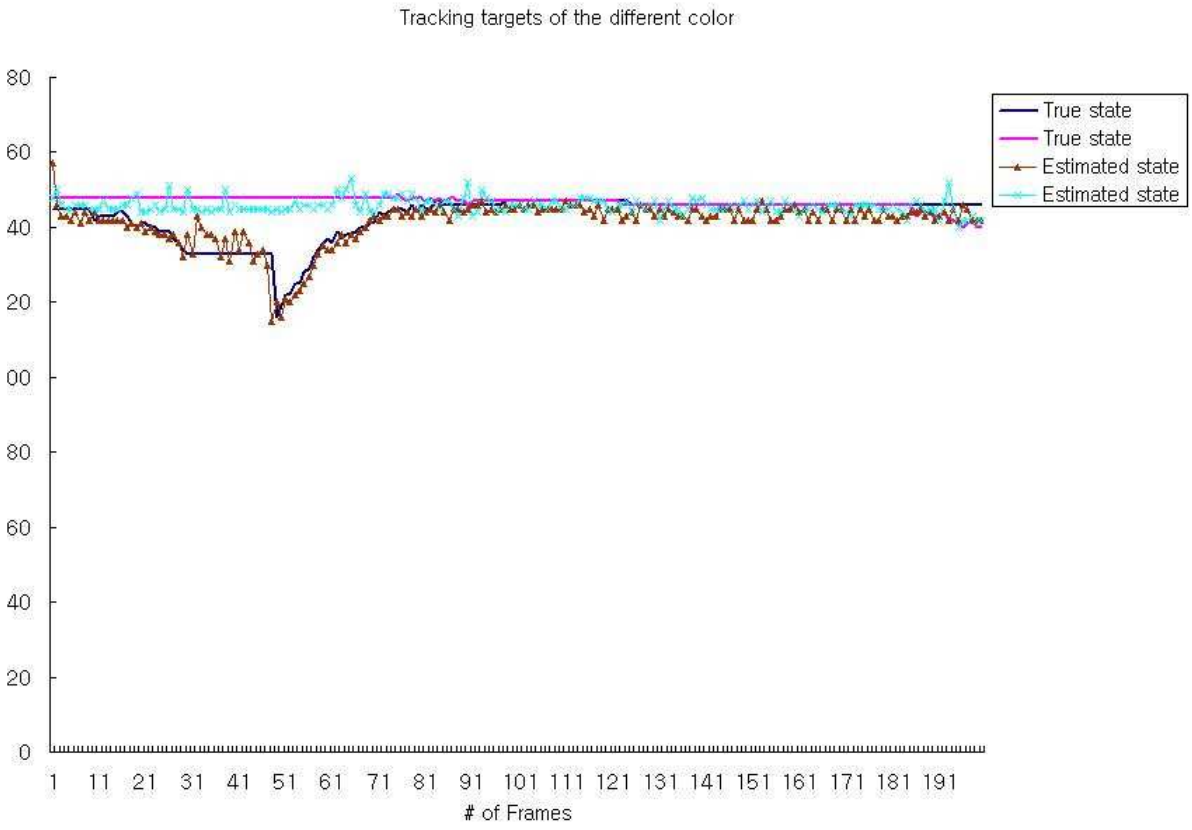
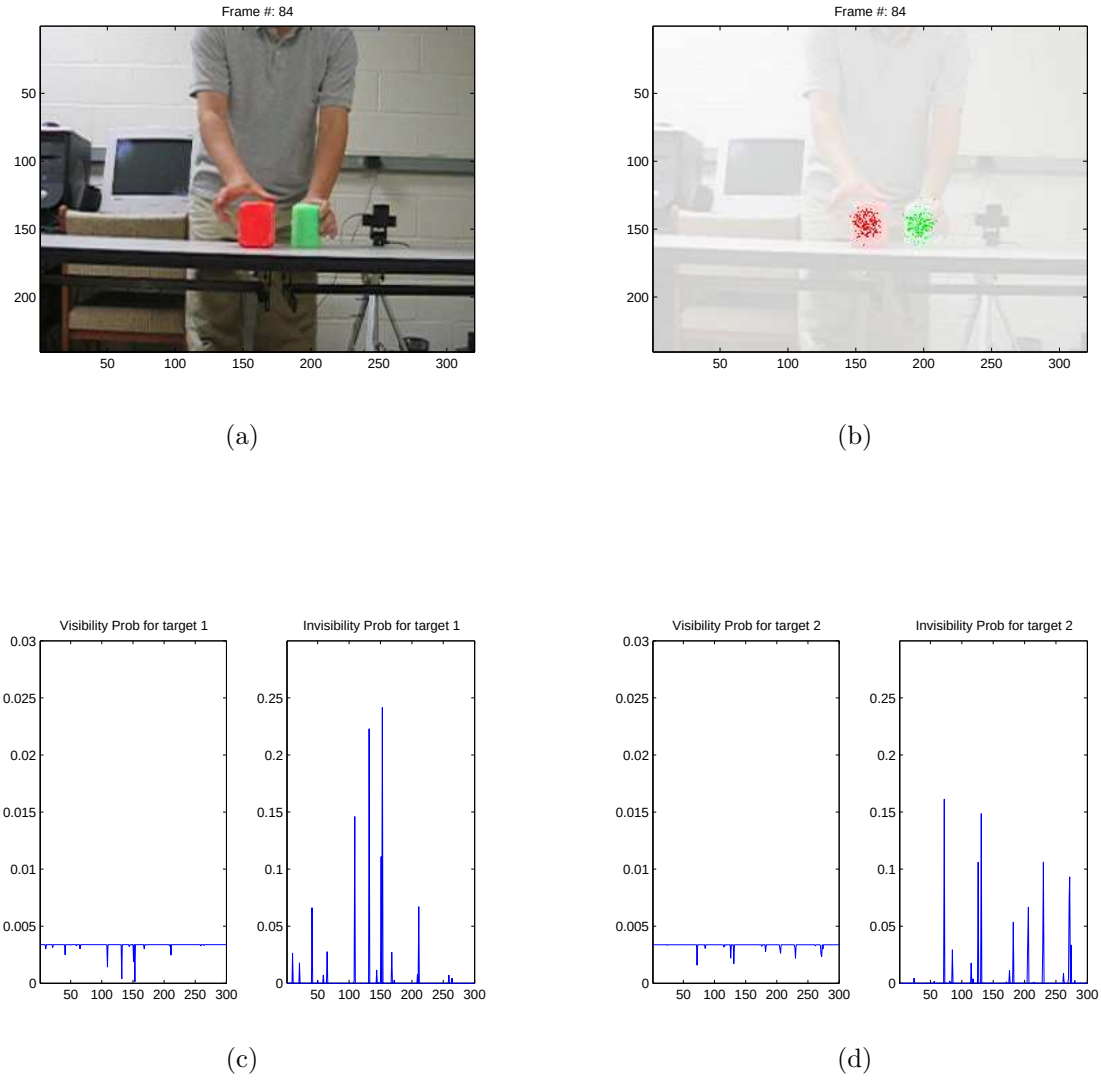


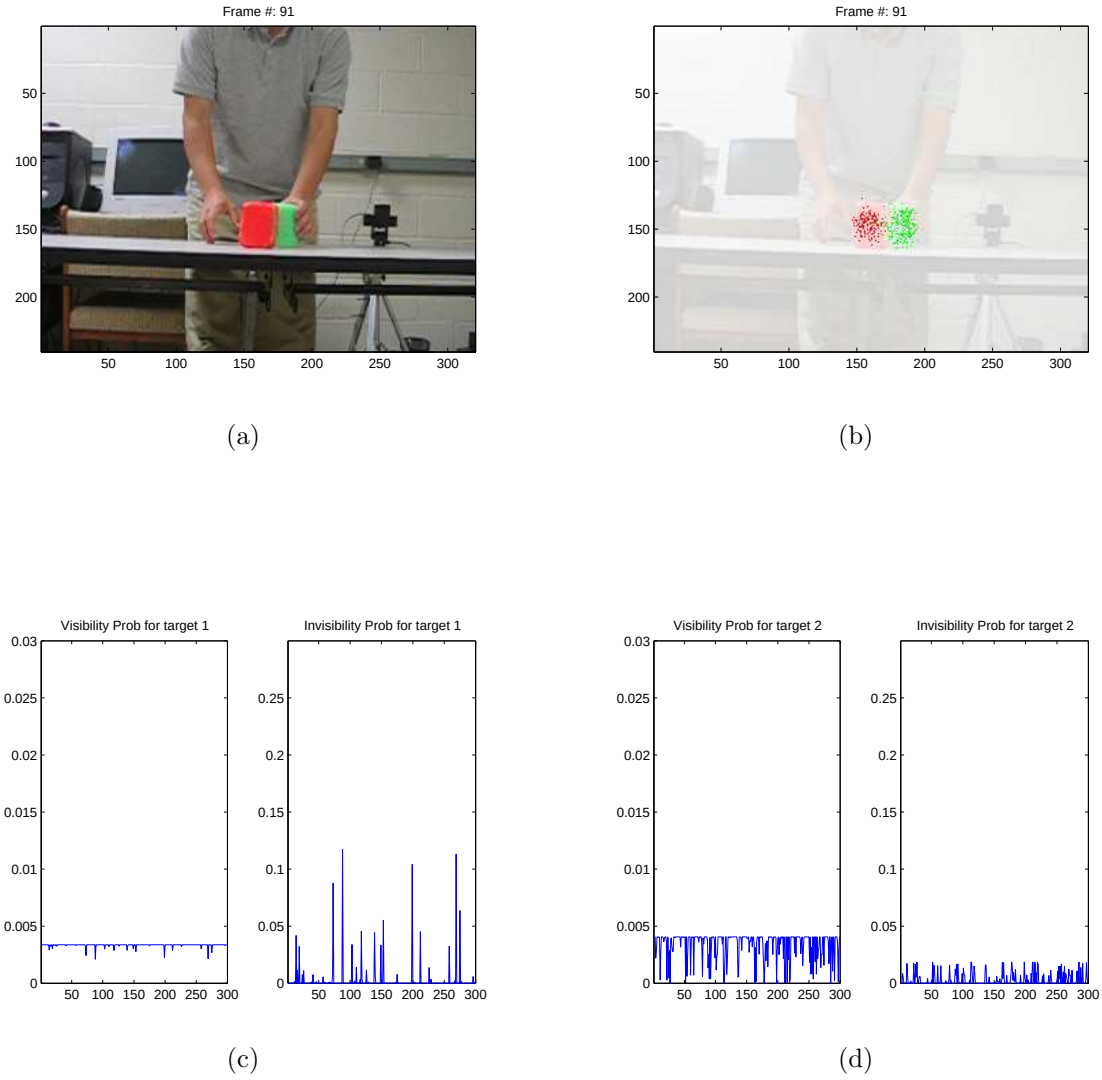
Figure 5.9. Tracking two targets in the Y axis.



(a) (b)

(c) (d)

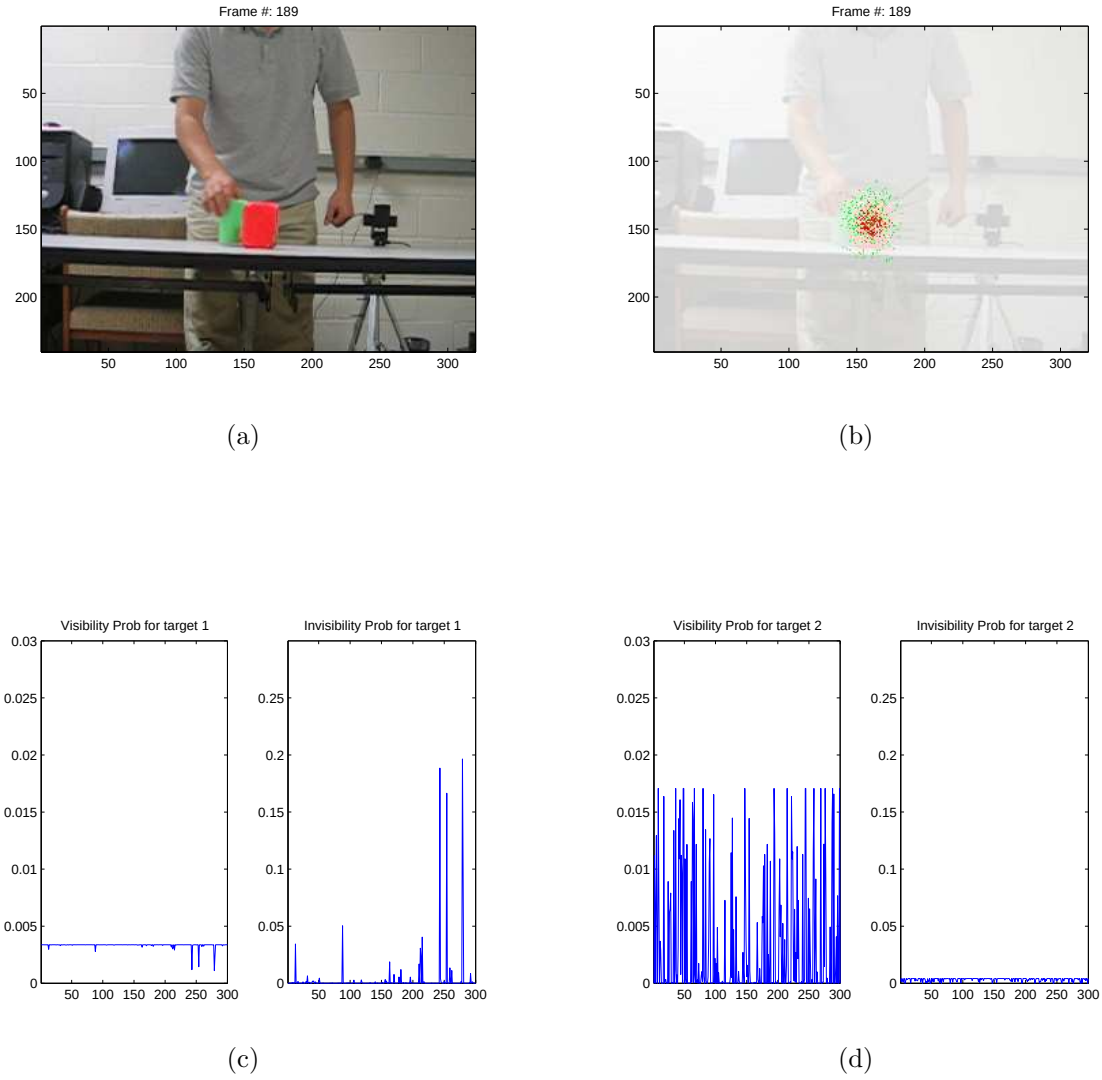
Figure 5.10. (a) and (b) are the image before occlusion happens at frame 84 (c) Visibility and Invisibility density for red target at frame 84 (d) Visibility and Invisibility density for green target at frame 84.



(a) (b)

(c) (d)

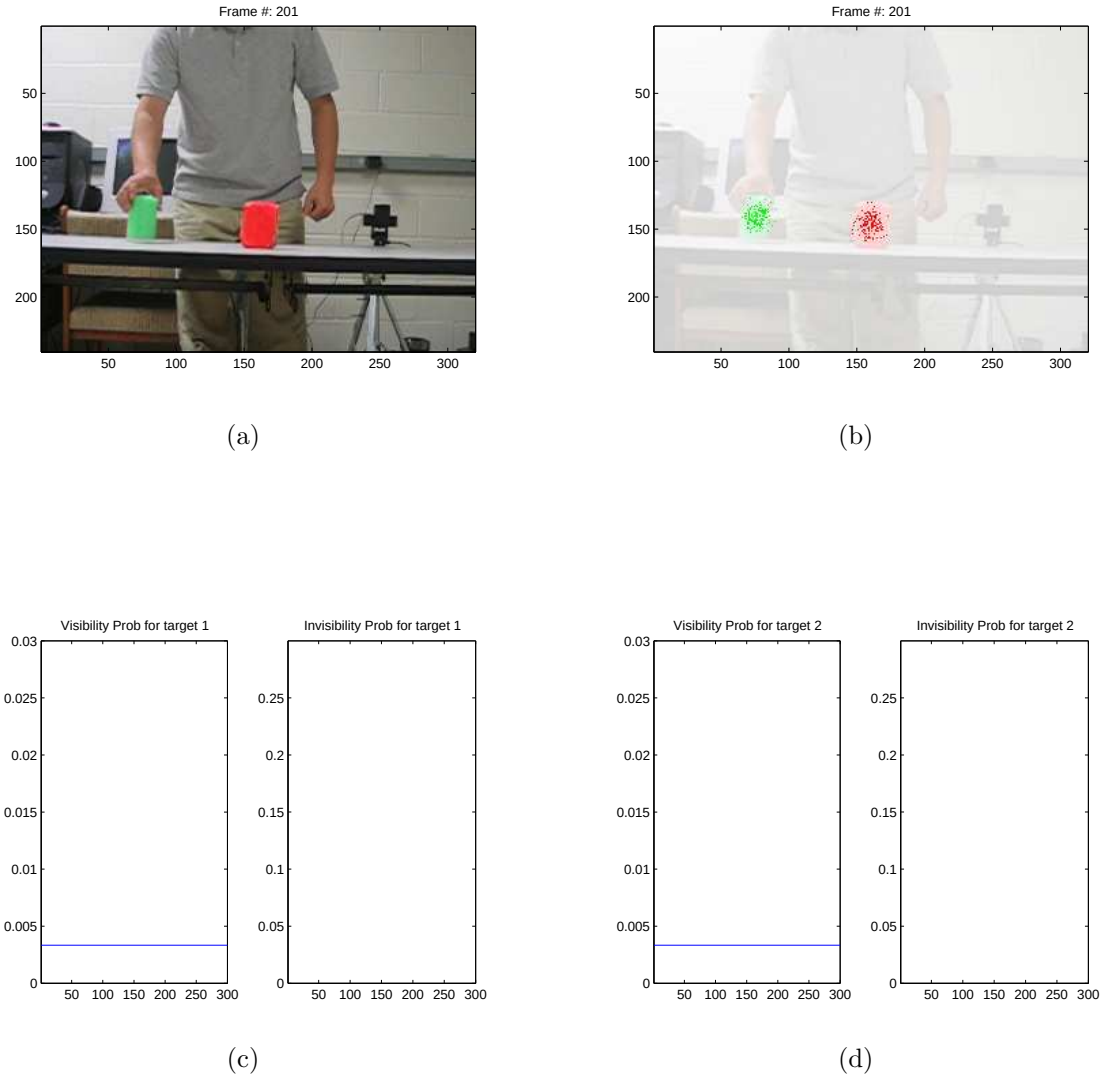
Figure 5.11. (a) and (b) are the image while occlusion happens at frame 91 (c) Visibility and Invisibility density for red target at frame 91 (d) Visibility and Invisibility density for green target at frame 91.



(a) (b)

(c) (d)

Figure 5.12. (a) and (b) are the image while occlusion happens at frame 189 (c) Visibility and Invisibility density for red target at frame 189 (d) Visibility and Invisibility density for green target at frame 189.



(a) (b)

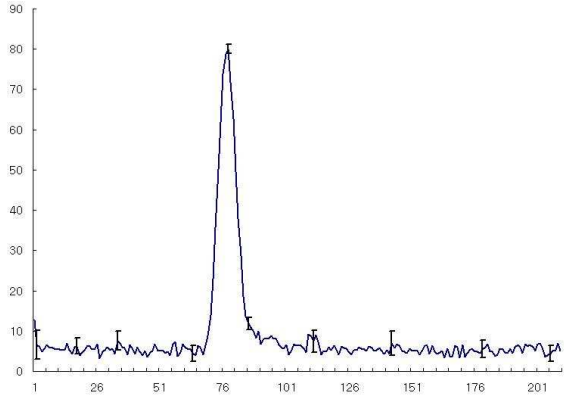
(c) (d)

Figure 5.13. (a) and (b) are the image after occlusion happens at frame 201 (c) Visibility and Invisibility density for red target at frame 201 (d) Visibility and Invisibility density for green target at frame 201.

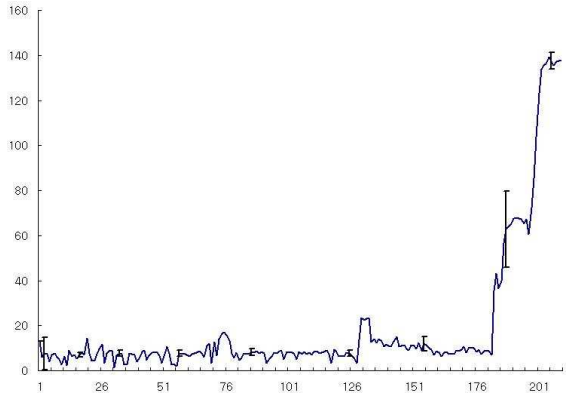
5.3.2.2 Tracking Three Targets Of The Different Colors

Figure 5.14 describes mean errors of tracking three targets using the proposed multi-target Particle filter. One thing to note here is that peaks represent occlusion situations or target disappearance such that target 1 and target 2 in Figure 5.14 has peaks around frame 80 and 100 and the mean error of target 3 increases at the end because this target starts to disappear. In short, this figure shows that the proposed method is capable of tracking three targets under occlusion situations. The higher error during occlusion can be attributed to the particles explaining out behind the occluding object due to the lack of target-specific observation.

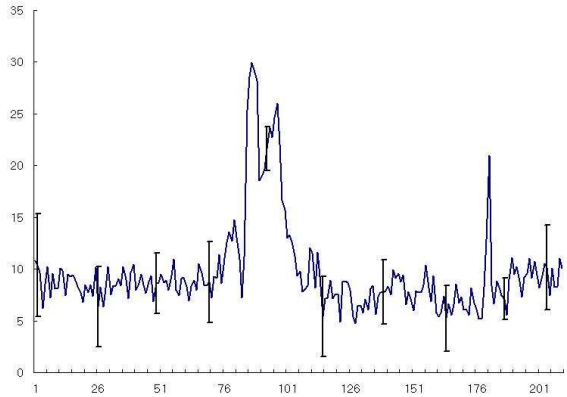
Figure 5.15 and Figure 5.16 show behaviors of three targets in the X and the Y direction. We can look at an occlusion of the three targets from frame 133 to frame 177 in Figure 5.15. In Figure 5.17, the two targets start occluding each other and invisibility probability density increase. Figure 5.19 and Figure 5.20 show two consecutive sequence of images with the three targets occluding each other and its corresponding density figures show the interpretation of the occlusion by increasing its invisibility probability or visibility probability. The particles of occluded targets stay at the occluding target. The important thing to note is that the number of the targets does not influence the behavior of occluded particles.



(a) Target 1



(b) Target 2



(c) Target 3

Figure 5.14. Mean of error in Cartesian distance between estimated state and true state and error bars using distributed multi-target Particle filter.

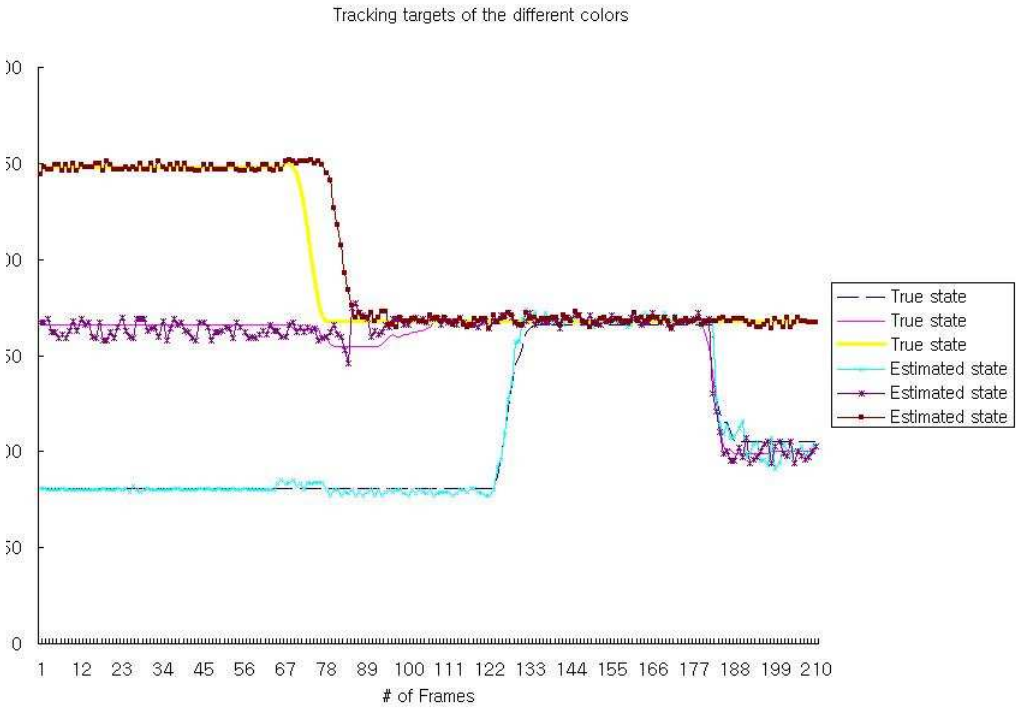


Figure 5.15. Tracking two targets in the X axis.

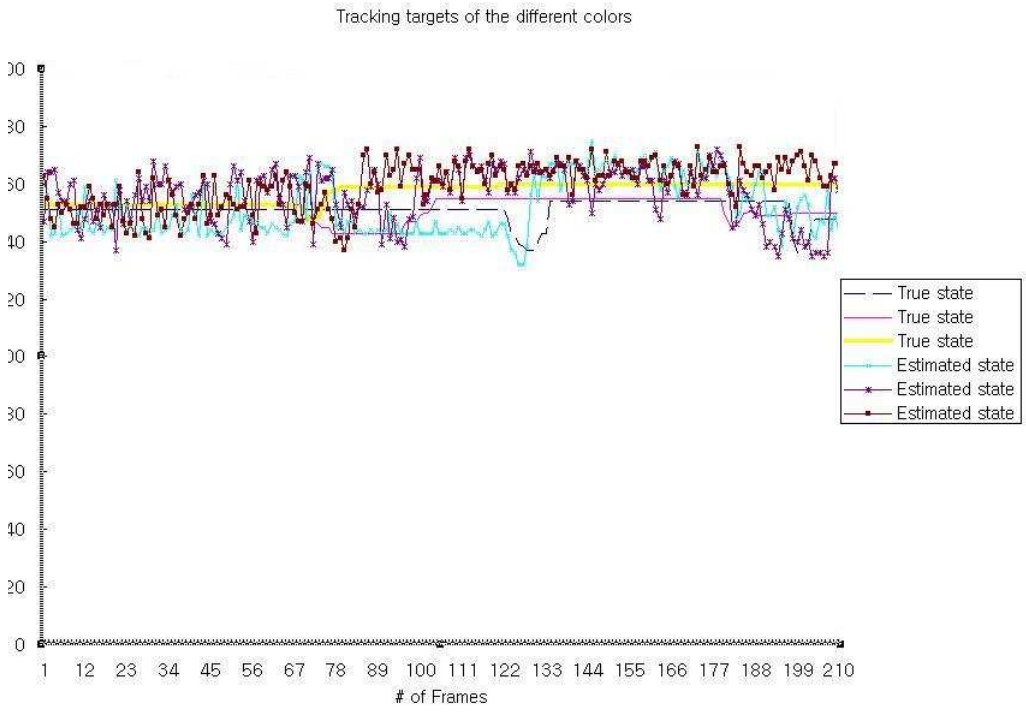


Figure 5.16. Tracking two targets in the Y axis.

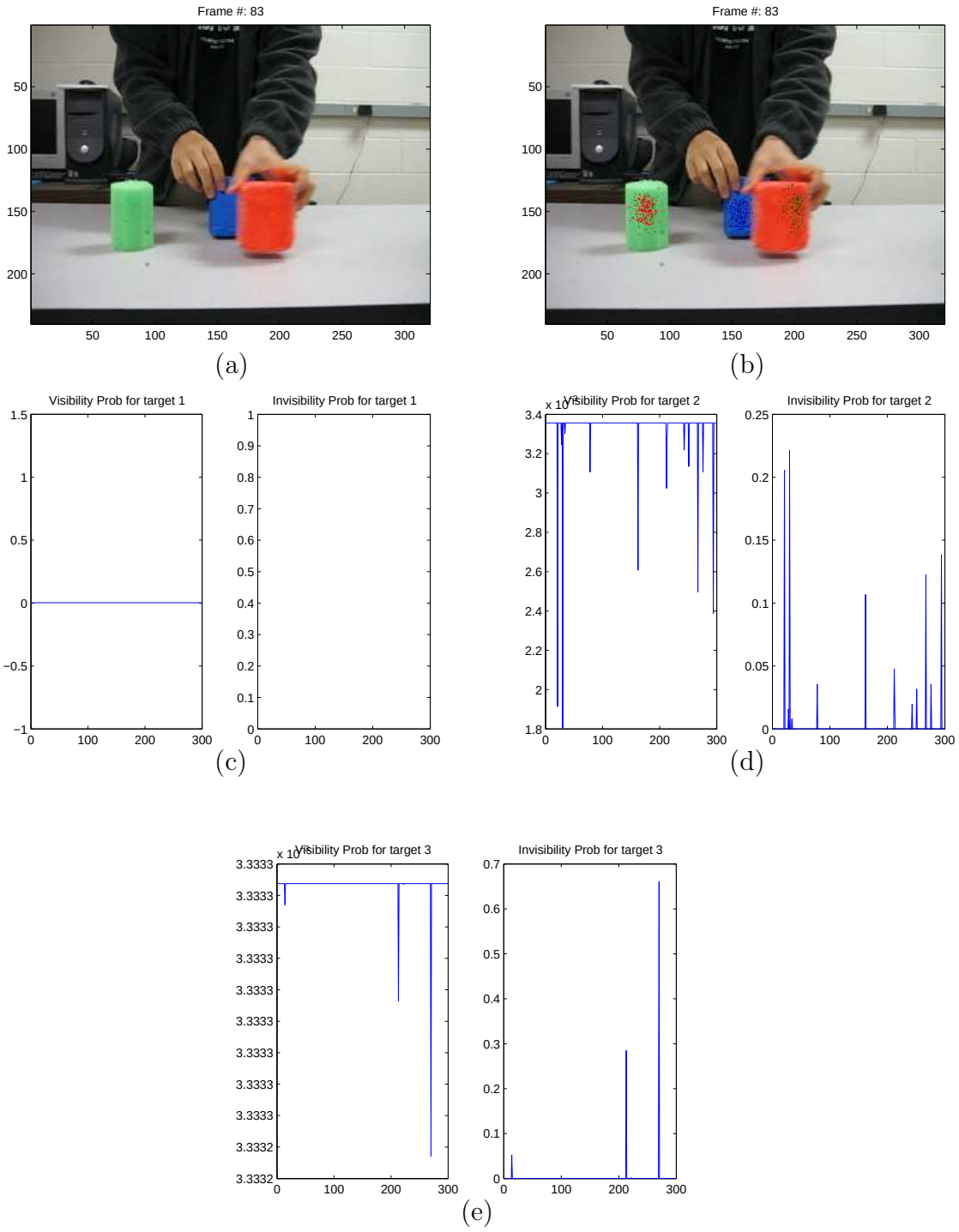


Figure 5.17. (a) and (b) Images at frame 83. (c) Visibility and Invisibility density of green target. (d) Visibility and Invisibility density of red target. (e) Visibility and Invisibility density of blue target.

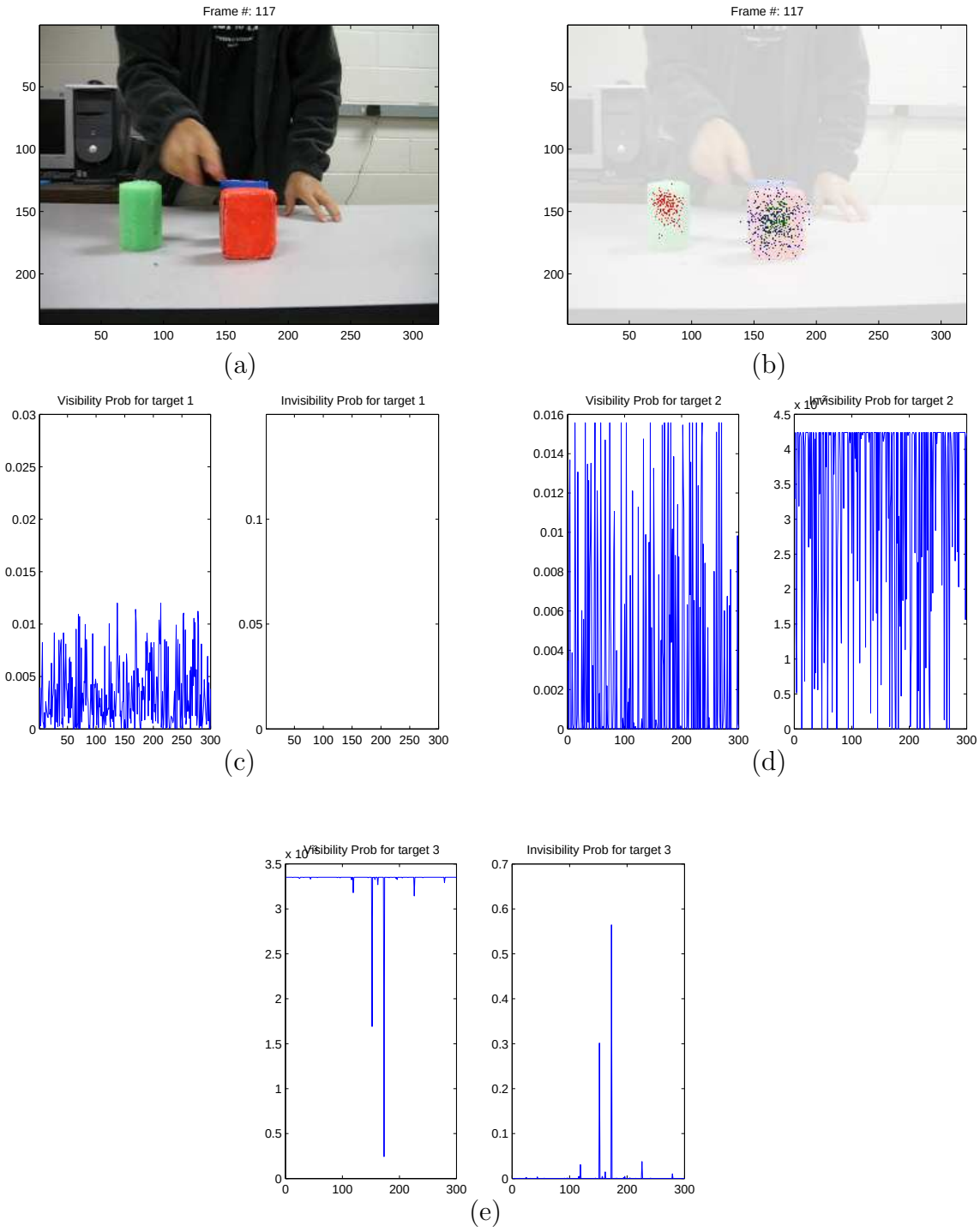


Figure 5.18. (a) and (b) Images at frame 117. (c) Visibility and Invisibility density of green target. (d) Visibility and Invisibility density of red target. (e) Visibility and Invisibility density of blue target.

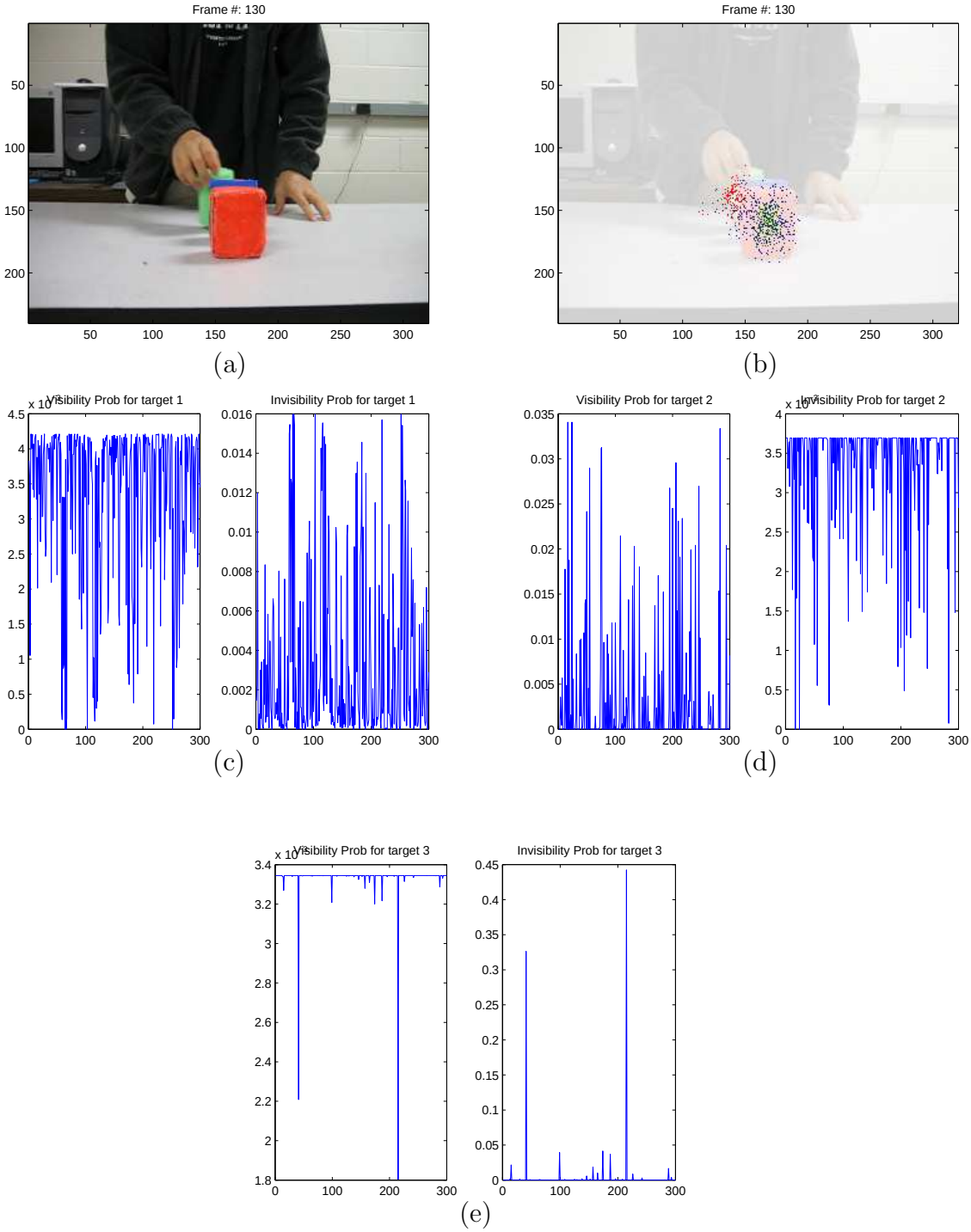


Figure 5.19. (a) and (b) Image at frame 130. (c) Visibility and Invisibility density of green target. (d) Visibility and Invisibility density of red target. (e) Visibility and Invisibility density of blue target.

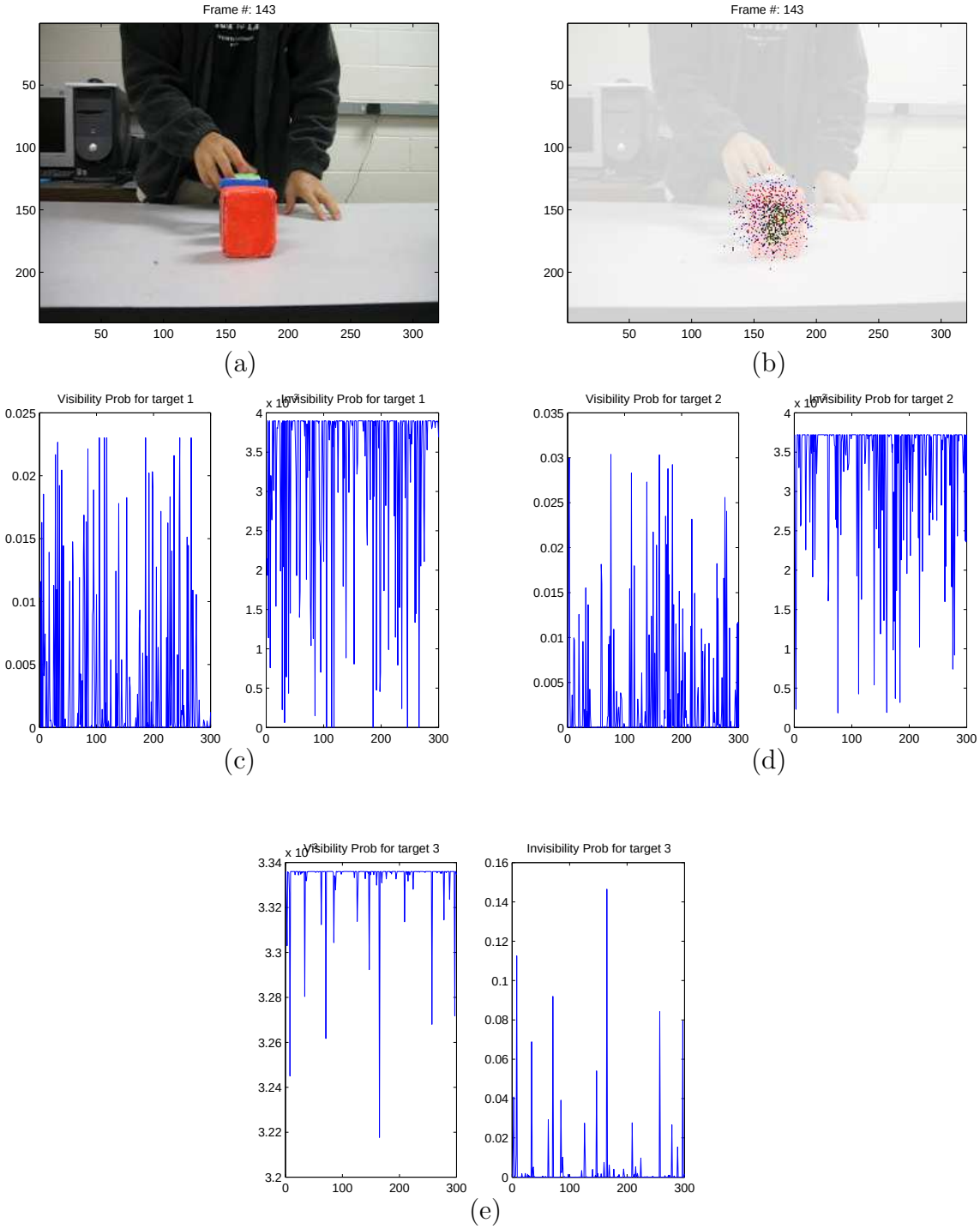


Figure 5.20. (a) and (b) Image at frame 143. (c) Visibility and Invisibility density of green target. (d) Visibility and Invisibility density of red target. (e) Visibility and Invisibility density of blue target.

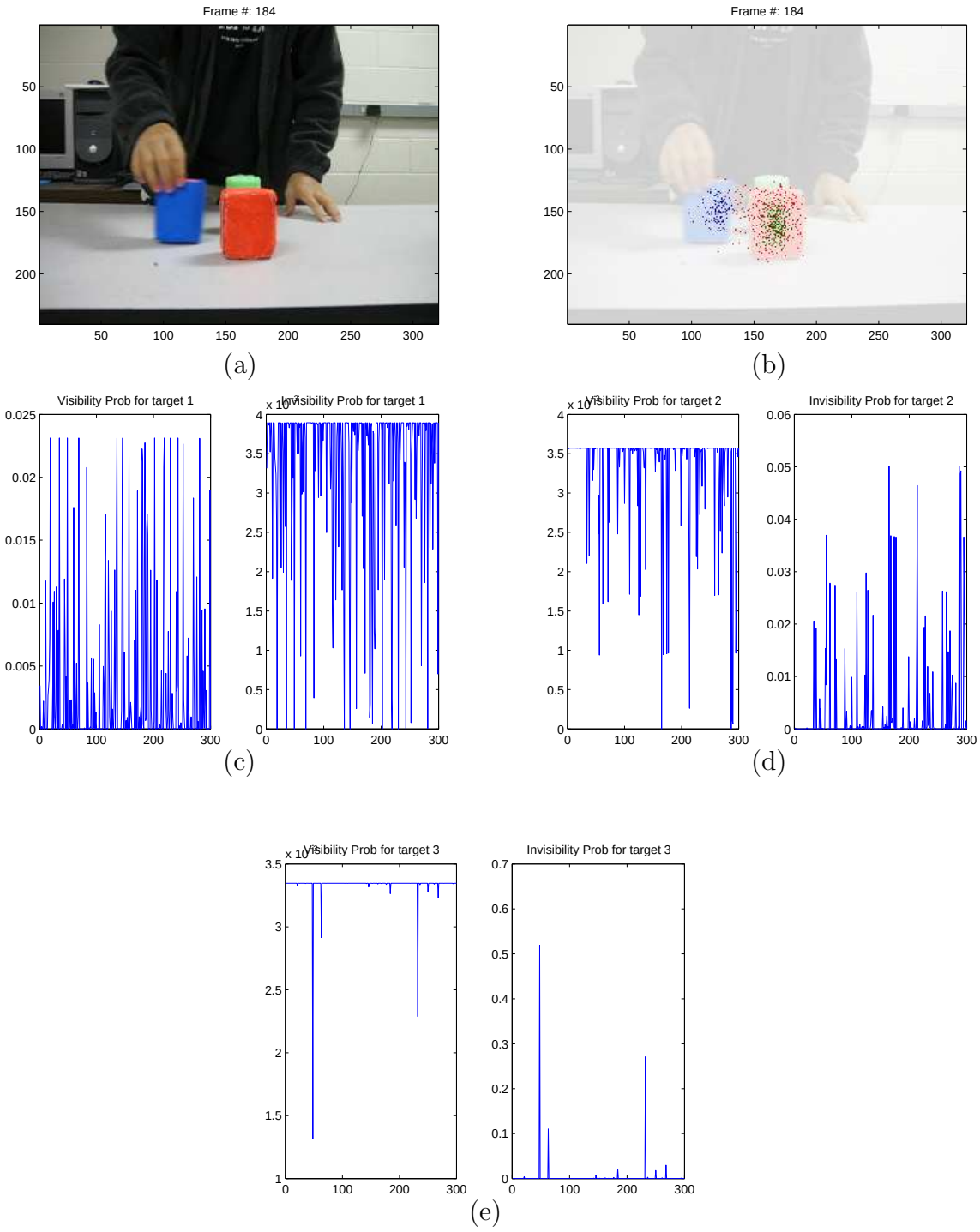


Figure 5.21. (a) and (b) Image at frame 184. (c) Visibility and Invisibility density of green target. (d) Visibility and Invisibility density of red target. (e) Visibility and Invisibility density of blue target.

5.3.2.3 Filter Deletion and Creation

The previous subsections demonstrate tractability even in the increasing number of targets and occlusion situations. This subsection demonstrates appearance and disappearance of targets by constructing and destroying a filter. In Figure 5.22, particle weights of Target 1 decreased as the target starts to disappear below the desk at frame 32 and a new target appears from the target 2 (plain line). The particle weights for the new target increase as the filter is getting strong measurement supports. The important element to note here is that dots located throughout the image represent additional filter (background filter) and play a role in surveilling a new target that might enter the scene. Figure 5.23 and Figure 5.24 show sequences of images where one target appears from the foreground target. We used the same color targets because it is generally easier for a hidden target to be detected if it has a the different color. As shown in the image figures, the background filter detects the hidden target 3 frames after the hidden target appears from behind the foreground target. In addition, Figure 5.22 also shows that the particle weight distribution (target 1) is destroyed around frame 38 and new a particle weight distribution is created around frame 59.

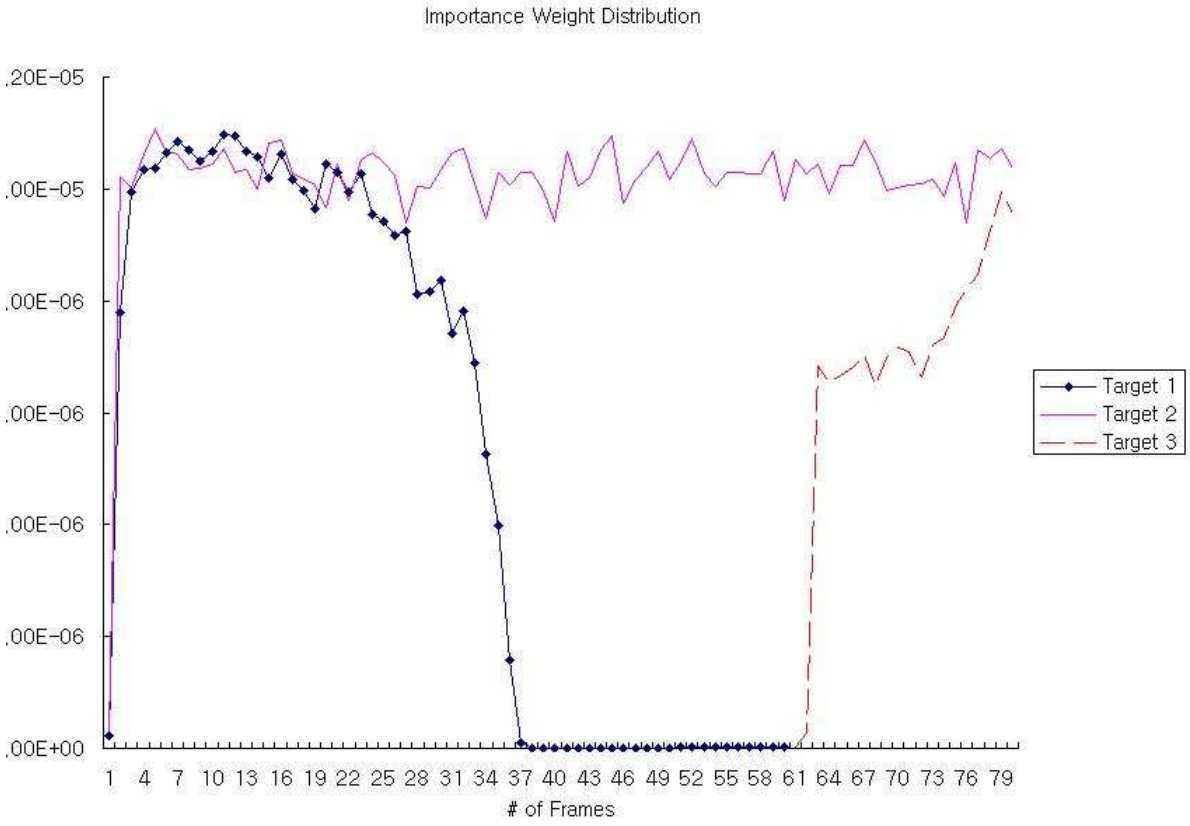


Figure 5.22. Importance Weight Distribution for three targets.

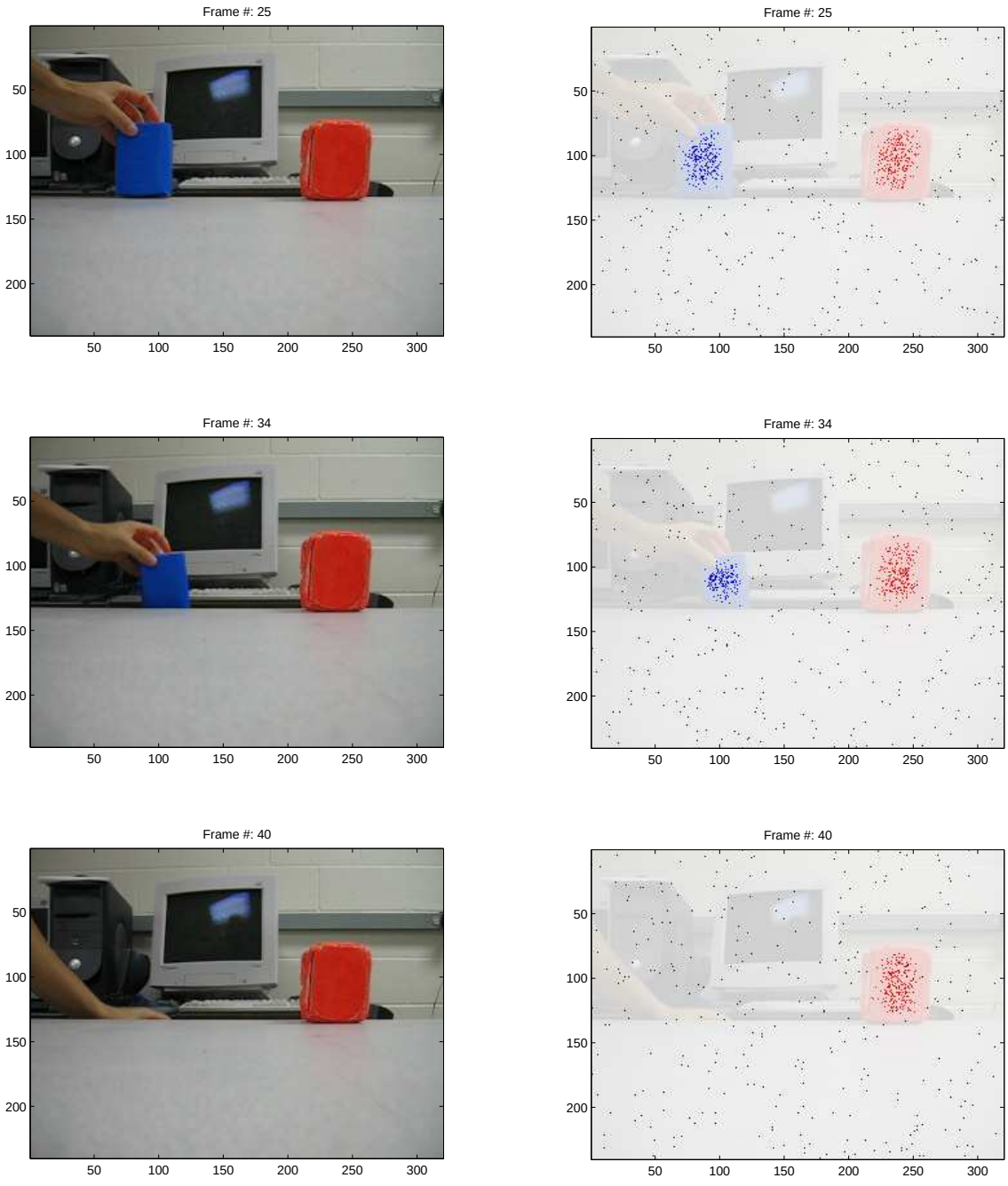


Figure 5.23. Visual tracking results on target deletion.

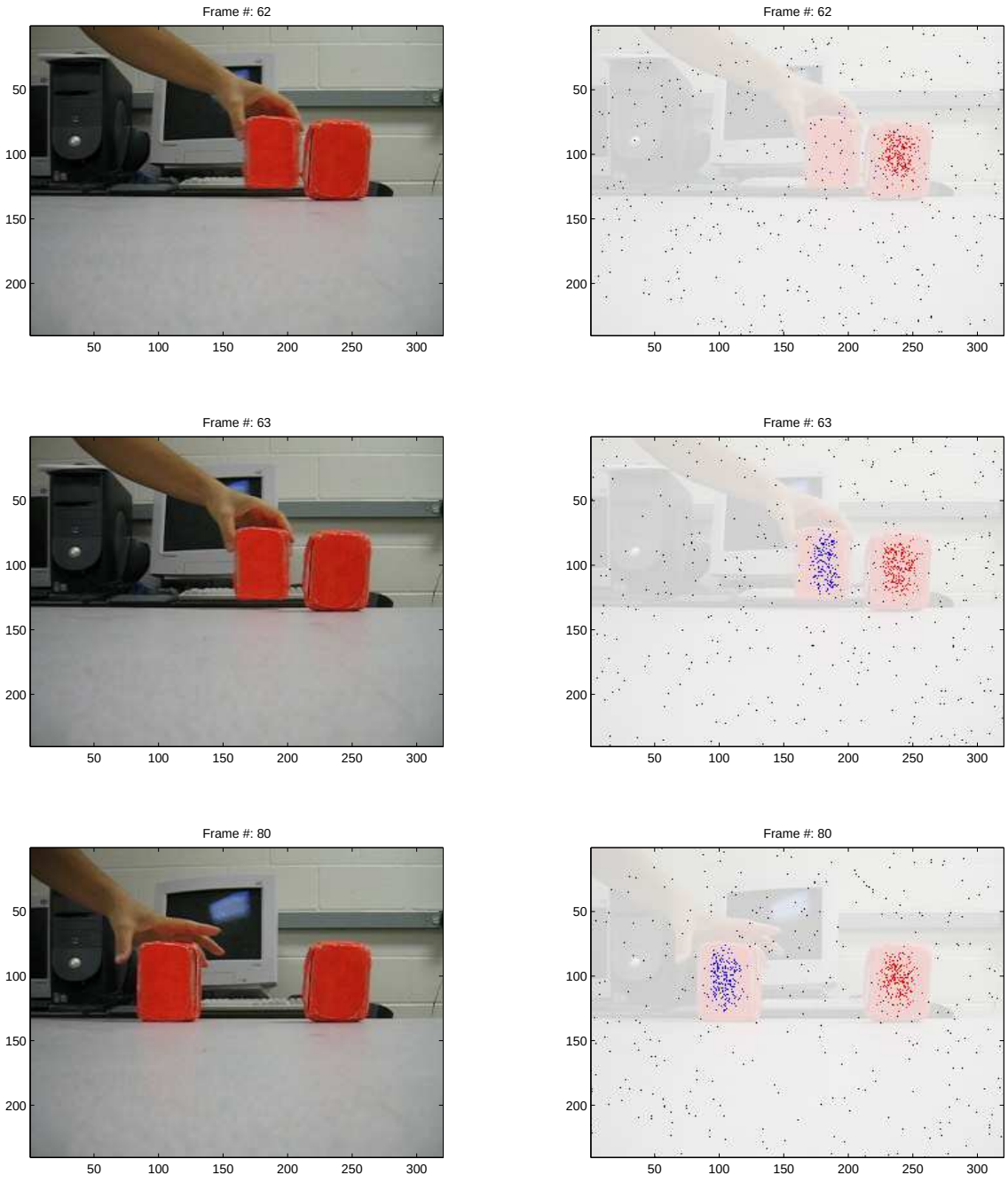
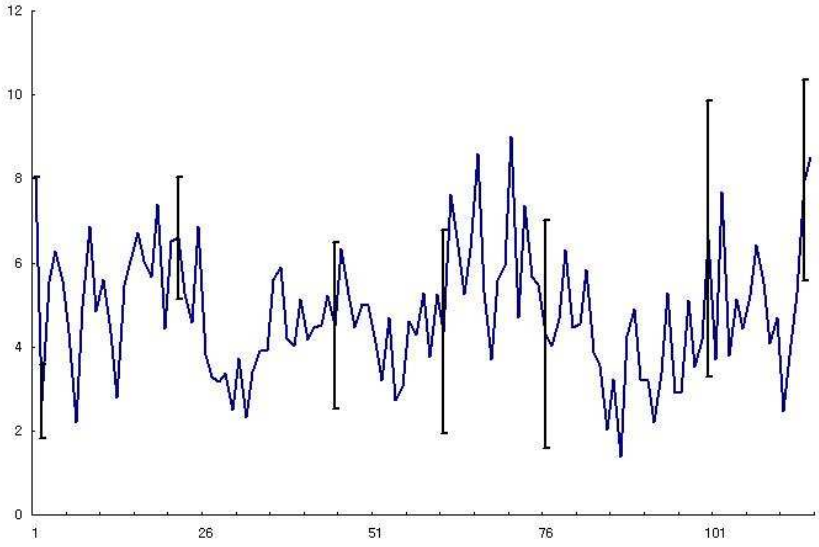


Figure 5.24. Visual tracking results on target creation.

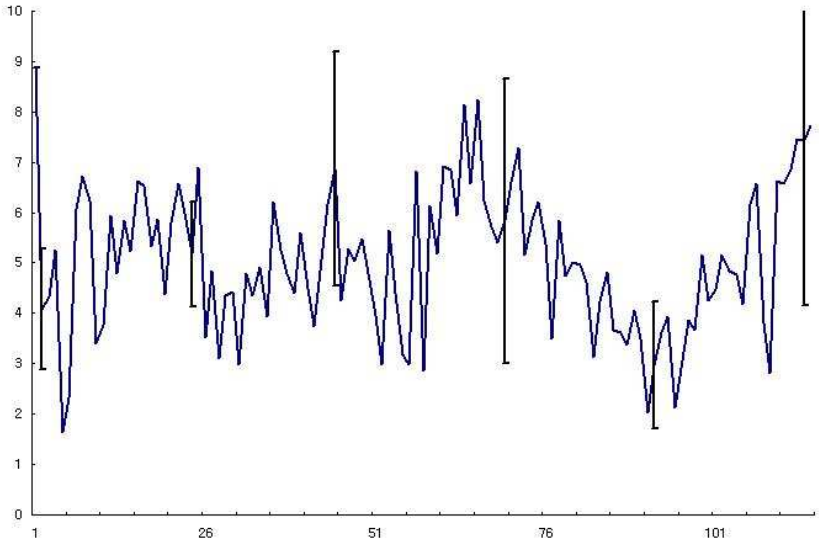
5.3.2.4 Experiment Summaries

Figure 5.25 shows the precision of tracking one target using the standard Particle filter and the proposed Particle filter under no collision and target disappearance. The mean errors in both figures are similar to each other and their standard deviations are similar as well. Figure 5.1 shows average of estimated states and their standard deviations as error bars for tracking two targets and three targets after multiple runs using standard Particle filter. Figure 5.7 shows the same information as Figure 5.1 but using the proposed multi-target Particle filter. The results of both figures indicate that the proposed multi-target Particle filter is capable of tracking all the targets by maintaining low standard deviation between a true state and an estimated state, whereas the standard Particle filter does not maintain its estimated state consistently and has a high standard deviation at the beginning on both figures because the standard Particle filter jumps into one target and later into the other target. Therefore, it is not suited for tracking multiple targets.

Figure 5.26 and Figure 5.14 describe evaluation of tracking three targets using the standard Particle filter and the proposed multi-target Particle filter. Similar to the results for tracking two targets, the proposed method is capable of tracking three targets under occlusion situations, whereas the standard Particle filter changes a tracking target when the targets are occluded as shown in Figure 5.26 (b) and (c) during frames 176 and 210.

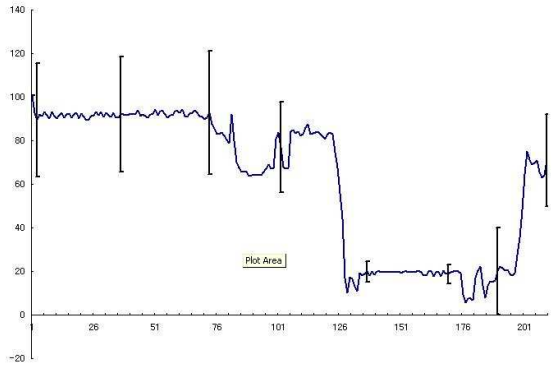


(a)

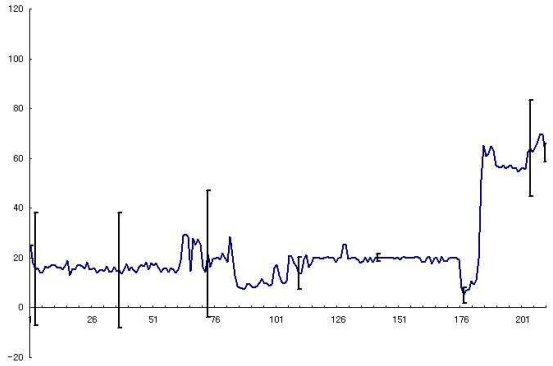


(b)

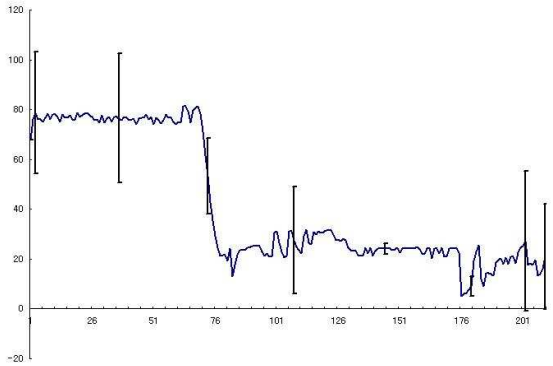
Figure 5.25. Mean of errors in Cartesian distance and error bars using the standard Particle filter (a) and multi-target Particle filter (b).



(a)



(b)



(c)

Figure 5.26. Mean of errors in Cartesian distance and error bars using standard Particle filter. (a), (b), and (c) are targets of the different colors and measurement supports are multiplication of each observation model.

CHAPTER 6

CONCLUSIONS

A primary contribution of this thesis is its demonstration of tracking multiple targets while maintaining a computational complexity that only adds a constant factor to a standard Particle filter. As the standard Particle filter as well as the mixture Particle filter have problems in multi-target tracking, we proposed an extended version of the Particle filter to remedy the problems while avoiding the complexity of a filter using a joint distribution model. The proposed Particle filter is well suited to explain a *joint* Particle distribution for visual tracking but it decreases the exponential complexity by maintaining a distributed filter representation to a complexity that adds only a constant factor to the standard Particle filters.

Versions of the multi-target particle filters described in Chapter 2 proposed different ways to make inferences about occlusions, but the proposed approach achieves approximation to the joint observation model by projecting from particle space to image space, while maintaining the complexity of the mixture model. The method effectively helps tracking multiple targets robustly even under occlusion situations. The proposed approach is evaluated through a number of experiments and demonstrated its precision in terms of tracking targets of different colors or the same colors. In addition, we developed a filter deletion and creation approach using the joint observation model so that the proposed approach is capable of tracking targets entering or disappearing the scene without influencing the complexity or biasing each target distribution.

Consequently, the proposed approach strengthens the joint Particle filter by approximating a new joint observation model to track multiple targets under the assumption that the target region corresponding to each particle is monolithic, each pixel only comes from one target, and all particles contribute equally to the particle observation without increasing its complexity by projecting between the particle space and the image space.

REFERENCES

[1] M. Isard and J. MacCormick, "Bramble: A bayesian multiple-blob tracker." in *ICCV*, 2001, pp. 34–41.

[2] N. Wiener and E. Hopf, "On a class of singular integral equations," 1931, p. 696.

[3] N. Wiener, *Extraplation, Interpolation and smoothing of Time Series, with Engineering Application.* United States of America: New York: Wiley, 1949.

[4] A. N. Kolmogorov, "Stationary sequences in hilbert sapces," p. 40, 1941.

[5] T. R. Bayes, "Essay towards solving a problem in the doctrine of chances," pp. 370–418, 1763.

[6] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory.* United States of America: New York: Wiley, 1998.

[7] M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework." in *ECCV (1)*, 1998, pp. 893–908.

[8] A. Doucet, "On sequential monte carlo sampling methods for bayesian filtering," 1998. [Online]. Available: http://citeseer.ist.psu.edu/doucet00sequential.html

[9] Z. Khan, T. R. Balch, and F. Dellaert, "An mcmc-based particle filter for tracking multiple interacting targets." in *ECCV (4)*, 2004, pp. 279–290.

[10] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining multi-modality through mixture tracking." in *ICCV*, 2003, pp. 1110–1116.

[11] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter.* United States of America: Artech House, 2004.

[12] S. Maskell, M. Rollason, N. Gordon, and D. Salmond, "Efficient particle filtering for multiple target tracking with application to tracking in structured images." *Image Vision Comput.*, vol. 21, no. 10, pp. 931–939, 2003.

[13] A. Blake and M. Isard, "The condensation algorithm - conditional density propagation and applications to visual tracking." in *NIPS*, 1996, pp. 361–367.

[14] S. L. Dockstader and A. M. Tekalp, "Tracking multiple objects in the presence of articulated and occluded motion." in *Workshop on Human Motion*, 2000, pp. 88–.

[15] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking." in *ECCV (1)*, 2002, pp. 661–675.

[16] H. Tao, H. S. Sawhney, and R. Kumar, "A sampling algorithm for tracking multiple objects." in *Workshop on Vision Algorithms*, 1999, pp. 53–68.

[17] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking." in *ECCV (1)*, 2004, pp. 28–39.

[18] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects." in *ICCV*, 1999, pp. 572–578.

[19] D. Tweed and A. Calway, "Tracking many objects using subordinated condensation." in *BMVC*, 2002.

[20] O. Lanz, "Occlusion robust tracking multiple objects," Online, August 2004, project PEACH.

[21] J. Vermaak, P. Pérez, M. Gangnet, and A. Blake, "Towards improved observation models for visual tracking: Selective adaptation." in *ECCV (1)*, 2002, pp. 645–660.

[22] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," 2002. [Online]. Available: http://citeseer.ist.psu.edu/maskell01tutorial.html

[23] C. Rasmussen, "Joint likelihood methods for mitigating visual tracking disturbances," 2001. [Online]. Available: citeseer.ist.psu.edu/rasmussen01joint.html

[24] A. S. Sabbi, "Object tracking in a stereo system using particle filter," The University of Texas at Arlington," Techinical Report, 2005.

[25] K. Nummiaro, E. Koller-Meier, and L. J. V. Gool, "An adaptive color-based particle filter." *Image Vision Comput.*, vol. 21, no. 1, pp. 99–110, 2003.

BIOGRAPHICAL STATEMENT

Hwangryol Ryu was born in Daejeon, Korea in 1975. He came to Hawaii in 2000 and received his B.S. in Computer Science from Brigham Young University-Hawaii in 2004. After graduation, he began his study pursuing master's degree in the department of Computer Science and Engineering at The University of Texas at Arlington in 2004. His research interests include Artificial Intelligence with focus on Object Tracking Methodology. He earned his M.S. in Computer Science and Engineering in May 2006.