ROTATION AND SCALING INVARIANT TARGET TRACKING USING PARTICLE FILTERS IN INFRARED IMAGE SEQUENCES

By

VIJAY VENKATARAMAN

Bachelor of Technology

Indian Institute of Technology, Madras

Chennai, India

2003

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2005

Thesis Approved:

Dr Guoliang Fan

Thesis Advisor

Dr Martin Hagan

Dr George Scheets

Dr A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my most sincere thanks to my advisor, Professor Guoliang Fan, for his guidance, support and encouragement throughout my M.S. study. I thank him for taking time to answer all my questions both in class and in the lab and providing a very encouraging research atmosphere. I believe that all the things I have learned from him will help me to develop my professional and personal skills in my future study. I am grateful to my advisory committee members: Professor Martin Hagan and Professor George Scheets, for accepting to be part of the committee and for reading the manuscript.

I would also like to thank in particular Li Tang for her efforts in helping me with my research work and providing constant support and useful suggestions for my work.

I would like to thank my friend S.Nithya for her help and encouragement to complete this work. For their friendship and fruitful discussions I would like to thank all members of Visual Communication and Image Processing Laboratory directed by Dr.Fan.

Most of all, I want to express my deepest thanks to my parents without whose support my MS study would not have been possible. I thank them for their love and support, without which i could not have achieved so much.

This research was supported by the DoD Army Research Laboratory under Grant W911NF-04-1-0221.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

Introduction

This thesis examines the problem of target detection and tracking in infrared (IR) image sequences. IR radiation lies in between the visible spectrum and the microwave spectrum. These radiations are characteristic of hot objects. All hot objects emit energy in the form of IR radiation. With the advent of sensors which can detect these radiations even from far away objects, this technique of IR imaging has become quite popular. IR imaging is possible irrespective of light conditions, as the principle of imaging is not based on visible light. IR imaging is a interesting technique in the sense that it is a passive imaging technique, passive implies that no energy is sent out from the camera but imaging is completely based on the infrared radiation from the object of interest (target). These properties of IR make it suitable for a number of applications such as in night vision, target tracking, fire fighting, medical diagnosis and the likewise.

The idea of not sending out a detecting signal (as in radar, sonar etc), but to use the energy of the target itself as signal makes infrared imaging very attractive to the military

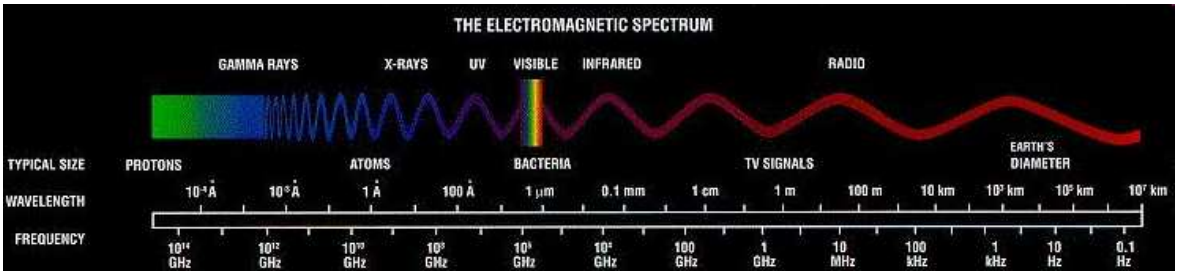


Figure 1.1: Electromagnetic (EM) spectrum

and a number of systems have been developed for target detection and tracking using information from the IR band of the EM spectrum. The thermal signatures of a number of military vehicles is shown in Fig. 1.2

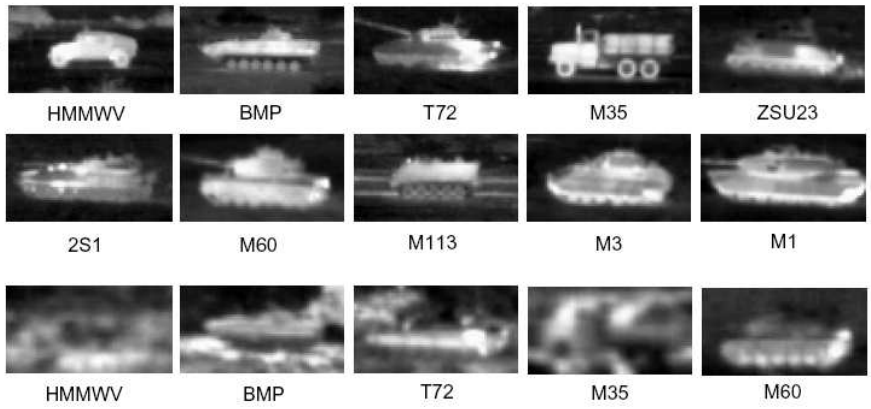


Figure 1.2: Thermal signatures of military vehicles

The infrared camera system used to capture the background information in our simulations is the Electrophysics PV320L2Z.An image of the camera is shown in Fig. 1.3

Figure 1.3: PV320 Infra red camera

In contrast to optical images, sequences obtained from an IR sensor have heavy clutter associated with them due to spurious heat reflectors in the scene, thus resulting in very poor Signal-to-Noise Ratio (SNR). The noise in infra-red images also has very strong correlation and cannot be considered as white noise or as simple gaussian noise. Some of the real world data acquired through the PV320 camera is shown in Fig. 1.4.

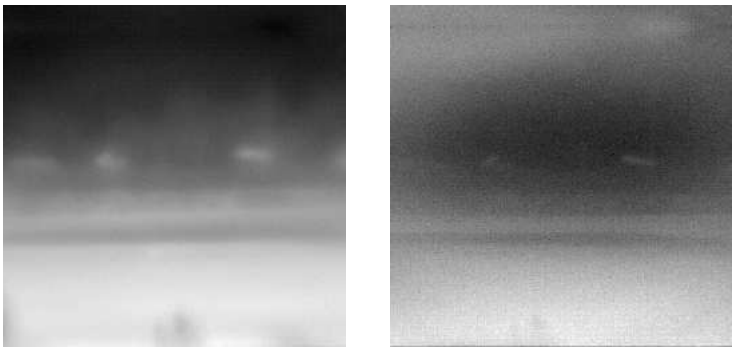


Figure 1.4: Long wave and mid wave band images of the same target

Intuitively one can infer from the images in Fig. 1.4 that the major problems in target tracking in infrared images will be poor target visibility resulting in a very low SNR, spurious heat emitters in the same scene of the target and in addition the non linear motion of the target and time varying aspects of the target. All these factors make target tracking in IR images more challenging and demands more research into the area.

1.1 Motivation

Traditional approaches found in literature [1] are based on the separation of detection and tracking problem. Most commonly the state space approach is used to track the target. The unknown state of a target estimated usually consisted of a set of kinematics components such as position, velocity or acceleration. Also these methods frequently do not use the raw sensor data but preprocessed outputs of preliminary detection systems.

Another traditional method is to use Kalman Filtering. The posterior distribution of the state of the target, which includes its position and velocity, is estimated recursively in time given the state transition and the measurement. Still it is inadequate in many cases because it is based on Gaussian densities which, being unimodal, cannot represent simultaneous alternative hypotheses [2].

Some recent approaches [3] have combined the detection and tracking into one framework by the use of particle filters, and have also use the raw data from the sensor itself.

The particle filtering method has been shown to out perform the Kalman-Bucy filtering approach. Particle Filters (PF) is a probability propagation model in signal processing which is effective for solving tracking problems, mainly as a result of the exponential growth of processor speed and memory capacity [4]. Tracking is modelled as a hidden state-space time series estimation problem in the Bayesian framework, which is solved using the sequential Monte Carlo (MC) estimation methods [5].

Previous researches on target detection/tracking with PF mostly deal with target having constant shape and intensity. Some works on multi-aspect target tracking assume that the target aspect state is defined on a finite, discrete-valued set [3]. Each index in this set is a pointer to one possible template model in the target aspect library, which is included as an additional variable in the state vector. In this case, the template library should include targets with all of the possible appearances resulting from rotation of its base template or zoom in/out effect of the camera. Moreover, the matrix of transition probabilities of the discrete-valued aspect cannot model a continuous aspect transition.

1.2 Contribution

We propose a new algorithm to overcome the limitations of the conventional multi-aspect target tracking methods using particle filters, in which the target aspect is modelled using a affine transformation model. The shape of the template at each time instant can be fully described by several affine parameters, which are augmented along with the position and velocity as new state variables. These parameters help us to simulate different target signatures without actually having to store all possible signatures of the target. With the new augmented state space, the algorithm can not only track the target but can also estimate its aspect, which can be valuable information in the task of target recognition.

1.3 Outline

This thesis is divided into five chapters. The first chapter consists of this introduction and it highlights the motivations and contributions of this work. Chapter two describes the state space model and state estimation methods, with special attention to the theory of particle filtering and its variations. Chapter three presents the problem of target tracking using particle filters, and our state space model. In particular it describes the system and observation model and the implementation of the target tacking algorithm. In chapter four we discuss the problem of multi aspect target tracking, herein we define the various affine parameters which we will use in our model. The chapter continues to contain simulation results and some discussion about the results. Finally chapter 5 contains the conclusions and ideas for further work along the same lines.

CHAPTER 2

State space model and state estimation

The tracking problem has always been dealt with in a state space form so in this chapter we will cover the underlying theory of state space estimation, conventional methods used for state estimation and the theory of particle filters for state estimation.

2.1 Introduction

Many problems require the estimation of the state of a system, which changes over time, using a sequence of noisy measurements made from the system. These measurements are related to the current state of the system. In this thesis our focus will be on discrete-time formulation of the problem and we will use difference equations to model the evolution of the system with time, and the measurements are also assumed to be available at discrete times.

The state-space approach relies on the state vector of a system. The state vector contains all the information required to describe the system or process investigation. For example, in the tracking problem, this information is often related to the kinematic characteristics of the target, including position, velocity and acceleration of the target as state vectors. The measurement vector represents (noisy) observations that are related to the state vector. The state space approach makes it very convenient to handle multivariate data and nonlinear/non-Gaussian processes

For us to analyze and make inference about a dynamic system we need at least two models: First, a model which describes the evolution of the system state with time (the system model) and, second, a model which relates the noisy measurements to the state (the

measurement model). We will assume that these models are available in their probabilistic form as the probabilistic state-space formulation and the requirement for the updating of information on receipt of new measurements are ideally suited for the Bayesian approach. This provides a rigorous general framework for state estimation problems.

In the Bayesian approach to state estimation, we attempt to construct the posterior probability density function (pdf) of the system state based on all available information, including the set of received measurements. Since this pdf contains all available statistical information about the state, it can be considered as the complete solution to the estimation problem. In principle we can obtain an optimal (with respect to any criterion) estimate of the state from the pdf. A measure of how accurate our estimate is may also be obtained from the pdf.

In some problems it may be possible to collect the observations over a period of time and then do a batch processing on the data for state estimation, but many real time systems require the data be processed in real time therefore an estimate is required every time a measurement is received. In this case, a recursive filter is a convenient solution. A recursive filtering approach implies that received observations can be processed sequentially as and when they are received rather than as a batch so that it is neither necessary to store the complete set of observations nor to reprocess existing data if a new measurement becomes available. Such a filter usually encompasses two stages: prediction and update. The prediction stage uses the system model to predict the state pdf forward from one measurement time to the next. Since the state is usually subject to unknown disturbances (modelled as random noise), the prediction step generally translates and deforms the state pdf. The update operation uses the latest measurement to modify the predicted pdf. This is achieved using Bayes theorem, which is the mechanism for updating knowledge about the system state knowing the extra information from new observation data.

In the following sections we will discuss the nonlinear tracking problem and its optimal Bayesian solution. When some conditions hold, this optimal solution becomes tractable

and can be computed using methods like Kalman filter and grid-based filter.

Often, the optimal solution is intractable because the constraints imposed by the above methods are not met. To solve these problems we have methods which provide several different approximation strategies to the optimal solution. These approaches include the extended Kalman filter, approximate grid-based filters, and particle filters.

2.2 Bayesian approach

Consider the state sequence $\{x_k, k \epsilon N\}$ of a target to evolve according to

$$x_k = f_k(x_{k-1}, v_{k-1}) \tag{2.1}$$

where $f_k : \Re^{n_x} \times \Re^{n_v} \rightarrow \Re^{n_x}$ is a nonlinear function of the states $x_{k-1}, \{v_{k-1}, k \epsilon N\}$ is an i.i.d. process noise sequence,n_x, n_v are dimensions of the state and process noise vectors, respectively and N is the set of natural numbers.

The objective of target tracking is to recursively estimate x_k from measurements

$$z_k = h_k(x_k, n_k) \tag{2.2}$$

where $h_k : \Re^{n_x} \times \Re^{n_n} \rightarrow \Re^{n_z}$ is a possibly nonlinear function,$\{n_k, k \epsilon N\}$ is an i.i.d. measurement noise sequence,n_z and n_n are dimensions of the measurement and measurement noise vectors, respectively. In particular, we try to find estimates of x_k based on the set of all available measurements, $z_{1:k} = \{z_i, i = 1, ..., k\}$ up to time k.

From a Bayesian perspective, the tracking problem is to recursively estimate the state x_k at time k, given the observation data $z_{1:k}$ up to time k. Therefore it is required to construct the pdf $p(x_k \mid z_{1:k})$. It is assumed that the initial pdf $p(x_0 \mid z_0) \equiv p(x_0)$ of the state vector (prior) is available (z_0 being the set of no measurements). Then, in principle, the pdf $p(x_k|z_{1:k})$ may be obtained, recursively, in two stages: prediction and update.

If we suppose that the required pdf $p(x_{k-1} \mid z_{1:k-1})$ at time k-1 is available, the prediction stage involves using the system model 2.1 to obtain the prior pdf of the states at time k using the Chapman-Kolmogorov equation

$$p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \tag{2.3}$$

At time step k, when a measurement z_k becomes available, this may be used to update the prior (update stage) using Bayes' rule

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \tag{2.4}$$

where the normalizing constant

$$p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k) p(x_k \mid z_{1:k-1}) dx_k \tag{2.5}$$

depends on the likelihood function $p(z_k \mid x_k)$ defined by the in 2.2 as the measurement model and the known statistics of n_k. In the update stage 2.4, the measurement z_k is used to modify the prior density to obtain the posterior density of the current state of the target.

The recurrence of relations 2.3 and 2.4 form the basis for the optimal Bayesian solution. This recursive propagation of the posterior density is only a conceptual solution and it usually cannot be determined analytically. Solutions exist in a restrictive set of cases, including the Kalman filter and grid-based filters.

2.3 Kalman filter (KF)

The Kalman filter assumes that the posterior density at every time step is Gaussian and, hence, can be fully characterized by a mean and covariance.

If $p(x_{k-1}|z_{1:k-1})$ is Gaussian, it can be proved that $p(x_k|z_{1:k})$ is also Gaussian, provided that certain assumptions hold:

v_{k-1} and n_k are drawn from Gaussian distributions of known parameters. $f_k(x_{k-1}, v_{k-1})$ is known and is a linear function of x_{k-1} and v_{k-1} and $h_k(x_k, n_k)$ is a known linear function of x_k and n_k. Therefore 2.1 and 2.2 can be rewritten as

$$x_k = F_k x_{k-1} + v_{k-1} \tag{2.6}$$

$$z_k = H_k x_k + n_k \tag{2.7}$$

F_k and H_k are known matrices defining the linear functions. The covariances of v_{k-1} and n_{k-1} are Q_{k-1} and R_k respectively. We consider the case where v_{k1} and n_k have zero mean and are statistically independent. Note that the system and measurement matrices F_k and H_k, as well as noise parameters Q_{k-1} and R_k , can be time variant.

The Kalman filter algorithm, which was derived in 2.3 and 2.4, can then be viewed as the recursion of the following relationships:

$$p(x_{k-1} \mid z_{1:k-1}) = N(x_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) \tag{2.8}$$

$$p(x_k \mid z_{1:k-1}) = N(x_k; m_{k|k-1}, P_{k|k-1}) \tag{2.9}$$

$$p(x_k \mid z_{1:k}) = N(x_k; m_{k|k}, Pk \mid k) \tag{2.10}$$

where

$$m_{k|k-1} = F_k m_{k-1|k-1} \tag{2.11}$$

$$P_{k|k-1} = Q_{k-1} + F_k P_{k-1|k-1} F_k^T \tag{2.12}$$

$$m_{k|k-1} = m_{k|k-1} + K_k(z_k - H_k m_{k|k-1}) \tag{2.13}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{2.14}$$

and where $N(x; m, P)$ is a Gaussian distribution with argument x, mean m, and covariance P, and

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{2.15}$$

$$K_k = P_{k|k-1}H_k^T + S_k^{-1} \tag{2.16}$$

are the covariance of the innovation term $z_k - H_k m_{k|k-1}$, and the Kalman gain, respectively.

This is the optimal solution to the tracking problem-if the highly restrictive assumptions of linearity of the system model and gaussian posterior hold. This means that no algorithm can ever do better than a Kalman filter in this linear Gaussian environment. At this point we would like to point out that it is possible to derive the same results using a least squares (LS) argument [6]. All the distributions will be then described by their means and covariances, and the algorithm remains unaltered, but are not constrained to be Gaussian distributions anymore. Assuming the means and covariances to be unbiased and consistent, the filter then optimally derives the mean and covariance of the posterior distribution. However, this posterior is not necessarily Gaussian and therefore if we define optimality as being the ability of an algorithm to calculate the posterior, the filter is then not certain to be optimal.

In many situations of interest, the assumptions made above do not hold. So The Kalman filter cannot be used as described.

2.4 Extended Kalman filter (EKF)

If 2.1 and 2.2 cannot be written in the form of 2.6 and 2.7 because the functions are nonlinear, then in a number of cases local linearization of the equations may be a sufficient description of the nonlinearity. The EKF works based on this linearization step and the fact that $p(x_k|z_{1:k})$ is approximated by a Gaussian distribution.

$$p(x_{k-1} \mid z_{1:k-1}) \approx N(x_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) \tag{2.17}$$

$$p(x_k \mid z_{1:k-1}) \approx N(x_k; m_{k|k-1}, P_{k|k-1}) \tag{2.18}$$

$$p(x_k \mid z_{1:k}) \approx N(x_k; m_{k|k}, Pk \mid k) \tag{2.19}$$

where

$$m_{k|k-1} = f_k(m_{k-1|k-1}) \tag{2.20}$$

$$P_{k|k-1} = Q_{k-1} + F_k P_{k-1|k-1} \hat{F_k^T} \tag{2.21}$$

$$m_{k|k-1} = m_{k|k-1} + K_k(z_k - h_k m_{k|k-1}) \tag{2.22}$$

$$P_{k|k} = P_{k|k-1} - K_k \hat{H}_k P_{k|k-1} \tag{2.23}$$

and where now $f_k(.)$ and $h_k(.)$ are nonlinear functions, and $\hat{F_k}(.)$ and $\hat{F_k}(.)$ are local linearizations of these nonlinear functions (i.e., matrices)

$$\hat{F_k}(.) = \frac{df_k(x)}{dx}|_{x=m_{k-1|k-1}} \tag{2.24}$$

$$\hat{H_k}(.) = \frac{dh_k(x)}{dx}|_{x=m_{k-1|k-1}} \tag{2.25}$$

$$S_k = \hat{H}_k P_{k|k-1} \hat{H_k^T} + R_k \tag{2.26}$$

$$K_k = P_{k|k-1} \hat{H_k^T} + S_k^{-1} \tag{2.27}$$

The EKF uses the first term in a Taylor expansion of the nonlinear function. A higher order EKF that retains further terms in the Taylor expansion exists, but it is not used widely since it has additional computational complexity associated with it.

However, the EKF always approximates $p(x_k|z_{1:k})$ to be Gaussian distribution. However if the true density happens to be non-Gaussian (e.g., if it is bimodal or heavily skewed),

then a Gaussian can never describe it well enough. It is in such cases that approximate grid-based filters and particle filters will give us an improvement in performance in comparison to that of an EKF [6].

2.5 Particle filtering methods

2.5.1 Sequential importance sampling (SIS) Algorithm

The sequential importance sampling (SIS) algorithm is a Monte Carlo (MC) method that forms the basis for most sequential MC filters developed over the past few years. This sequential MC (SMC) approach is also known as bootstrap filtering [5], the condensation algorithm [7] or particle filtering [8]. It is a technique for implementing a recursive Bayesian filter using Monte Carlo simulations.

The main idea in these methods is to represent the required posterior density function by a set of random samples with weights associated with each sample and to compute posterior estimates based on these samples and weights. As the number of samples becomes keeps increasing, this characterization using samples and weights becomes an equivalent representation to the usual functional description of the posterior pdf, and the SIS filter's estimate approaches the optimal Bayesian estimate.

In order to further develop the details of the particle filtering method, let $\{x^i_{0:k}, w^i_k\}^{N_s}_{i=1}$ characterize the posterior pdf $p(x_{0:k}|z_{1:k})$, where $\{x^i_{0:k}, i = 0, ..., N_s\}$, is a set of support points with associated weights $\{w^i_k, i = 1, ..., N_s\}$, and $x_{0:k} = \{x_j, j = 0, ..., k\}$, is the set of all states up to time k. The weights are normalized such that $\sum_i w^i_k = 1$. Then, the posterior density at time instant k can be approximated as

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w^i_k \delta(x_{0:k} - x^i_{0:k}) \tag{2.28}$$

We therefore have a discrete weighted approximation to the true posterior,$p(x_{0:k}|z_{1:k})$ where weights are chosen using the principle of importance sampling [9], [10]. This prin-

ciple is based on the following. "Suppose $p(x) \propto \pi(x)$ is a probability density from which it is difficult to draw samples but for which $\pi(x)$ can be evaluated. In addition, let $x^i \sim q(x), i = 1, ..., N_s$, be samples that are easily generated from a proposal $q(.)$ called an importance density. Then, a weighted approximation to the density $p(.)$ is given by

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i) \tag{2.29}$$

where

$$w^i = \frac{\pi(x^i)}{q(x^i)} \tag{2.30}$$

is the normalized weight of the i'th particle.

Therefore, if the samples $x^i_{0:k}$ were drawn from an importance density $q(x_{0:k}|z_{1:k})$, then the weights in 2.28 are defined 2.30 by to be

$$w^i_k \propto \frac{p(x^i_{0:k}|z_{1:k})}{q(x^i_{0:k}|z_{1:k})} \tag{2.31}$$

At each iteration, one has samples constituting an approximation to $p(x_{0:k-1}|z_{1:k-1})$ and want to approximate $p(x_{0:k}|z_{1:k})$ with a new set of samples. If the importance density is chosen to factorize such that

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}|z_{1:k-1}) \tag{2.32}$$

then one can obtain samples $x^i_{0:k} \sim q(x_{0:k}|z_{1:k})$by augmenting each of the existing samples $x^i_{0:k-1} \sim q(x_{0:k-1}|z_{1:k-1})$ with the new state $x^i_k \sim q(x_k|x_{0:k-1}, z_{1:k})$.
To derive the weight update equation, $p(x_{0:k}|z_{1:k})$ is first expressed in terms of $p(x_{0:k-1}|z_{1:k-1})$, $p(z_k|x_k)$, and $p(x_k|x_{k-1})$. Note that 2.4 can be derived by integrating 2.33

$$
\begin{aligned}
p(x_{0:k}|z_{1:k}) &= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_k|x_{0:k-1}, z_{1:k-1})}{p(z_k|z_{1:k-1})} \times p(x_{0:k}|z_{1:k-1}) \qquad (2.33) \\
&= \frac{p(z_k|x_k)p(x_k|x_{k-1})}{p(z_k|z_{1:k-1})} p(x_{0:k-1}|z_{1:k-1}) \\
&\propto p(z_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|z_{1:k-1}) \qquad (2.34)
\end{aligned}
$$

By substituting 2.32 and 2.34 into 2.31, the weight update equation can then be shown to be

$$
w_k^i \propto \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)p(x_{0:k-1}^i|z_{1:k-1})}{q(x_k^i|x_{0:k-1}^i, z_{1:k})q(x_{0:k-1}^i|z_{1:k-1})} = w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})} \qquad (2.35)
$$

Furthermore, if $q(x_k|x_{0:k-1}, z_{1:k}) = q(x_k|x_{k-1}, z_k)$, then the importance density becomes only dependent on x_{k-1} and z_k . This is particularly useful in the common case when only a filtered estimate of $p(x_k|z_{1:k})$ is required at each time step. In such scenarios, only x_k^i need be stored; therefore, one can discard the path $x_{0:k-1}^i$ and history of observations $z_{i:k-1}$. The modified weight is then

$$
w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})} \qquad (2.36)
$$

and the posterior filtered density can be approximated as

$$
p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \qquad (2.37)
$$

where the weights are defined in 2.36. It can be shown that as $N_s \to \infty$, the approximation 2.37 approaches the true posterior density $p(x_k|z_{1:k})$. The SIS algorithm thus consists of recursive propagation of the weights and support points as each measurement is obtained sequentially." [6]

Table 2.1: SIS particle filter

$[\{x_k^{j*}, w_k^j, i^j\}_{j=1}^{N_s}] = SIS[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, z_k]$

1. FOR $i = 1 : N_s$

   Draw $x_k^i \sim q(x_k|x_{k-1}^i, z_k)$

   Assign the particle a weight, w_k^i, according to 2.36

END FOR

2.5.2 Degeneracy problem and its solution

A common problem with the SIS particle filter is that after a few iterations al particles except a single one will have very small weights. This effect is reffered to as the degeneracy problem. It has also been shown in [10] that the variance of the weights can only increase over time, and thus, it is impossible to avoid the degeneracy phenomenon. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation of $p(x_k|z_{1:k})$is negligible or almost zero.

A suitable measure of degeneracy of the algorithm is the effective sample size and it is defined as

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s} 1 + var(w_k^{*i})} \tag{2.38}$$

where $w_k^{*i} = p(x_k^i|z_{1:k})/q(x_k^i|x_{k-1}^i, z_k)$ is referred to as the true weight but cannot be computed exactly. So an estimate $\hat{N_{eff}}$ of N_{eff} can be obtained as

$$\hat{N_{eff}} = \frac{1}{\sum_{i=1}^{N_s}(w_k^i)^2} \tag{2.39}$$

where w_k^i is the normalized weight obtained using 2.35. Always $N_{eff} \leq N_s$ and a small N_{eff} indicates severe degeneracy. A forceful approach to overcome this problem is to use large number of particles (very large N_s). This method is too impractical and so we use the following two methods 1. Good choice of importance density 2. Resampling

Choice of importance density

"The first method involves choosing the importance density $q(x_k^i|x_{k-1}^i, z_k)$ to minimize the $var(w_k^{*i})$ so that N_{eff} is maximized. There exists optimal importance density function which achieves this desired result and has been proved [10] to be

$$q(x_k^i|x_{k-1}^i, z_k)_{opt} = p(x_k^i|x_{k-1}^i, z_k) = \frac{p(z_k|x_k, x_{k-1}^i)p(x_k|x_{k-1}^i)}{p(z_k|x_{k-1}^i)} \tag{2.40}$$

substituting 2.40 in 2.36 yields

$$w_k^i \propto w_{k-1}^i p(z_k|x_{k-1}^i) \tag{2.41}$$

This choice of importance density is optimal since for a given x_{k-1}^i, w_k^i takes the same value, whatever sample is drawn from $q(x_k^i|x_{k-1}^i, z_k)_{opt}$, . Hence, conditional on x_{k-1}^i, $var(w_k^{*i}) = 0$. This is the variance of the different w_k^i resulting from different sampled x_k^i .

This optimal importance density suffers from two major drawbacks. It requires the ability to sample from $p(x_k^i|x_{k-1}^i, z_k)$ and to evaluate the integral over the new state. In the general case, it may not be straightforward to do either of these things. There are two cases when use of the optimal importance density is possible.

The first case is when x_k is a member of a finite set. In such cases, the integral in 2.41 becomes a sum, and sampling from $p(x_k^i|x_{k-1}^i, z_k)$ is possible. Analytic evaluation is possible for a second class of models for which is Gaussian [10]. This can occur if the dynamics are nonlinear and the measurements linear.

For many other models, such analytic evaluations are not possible. However, it is possible to construct suboptimal approximations to the optimal importance density by using local linearization techniques [10]. Such linearizations use an importance density that is a Gaussian approximation to $p(x_k^i|x_{k-1}^i, z_k)$. Another approach is to estimate a Gaussian approximation to $p(x_k^i|x_{k-1}^i, z_k)$ using the unscented transform [11]. But the general opinion is that the additional computational cost of using such an importance density is often more

than offset by a reduction in the number of samples required to achieve a certain level of performance.

Finally, it is often convenient to choose the importance density to be the prior

$$q(x_k^i|x_{k-1}^i, z_k) = p(x_k|x_{k-1}^i) \tag{2.42}$$

Substitution of 2.42 into 2.36 then yields

$$w_k^i \propto w_{k-1}^i p(z_k|x_k^i) \tag{2.43}$$

This would seem to be the most common choice of importance density since it is intuitive and simple to implement. However, there are a plethora of other densities that can be used, the choice is the crucial design step in the design of a particle filter." [6]

Resampling

"The second method by which the effects of degeneracy can be reduced is to use resampling whenever a significant degeneracy is observed (i.e., when N_{eff} falls below some threshold N_T). The basic idea of resampling is to eliminate particles that have small weights and to concentrate on particles with large weights. The resampling step involves generating a new set $\{x_k^{i*}\}_{i=1}^{N_s}$ by resampling (with replacement) N_s times from an approximate discrete representation of $p(x_k|z_{1:k})$ given by

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \tag{2.44}$$

so that $p(x_k^{i*} = x_k^j) = w_k^j$. The resulting sample is in fact an i.i.d. sample from the discrete density 2.44, therefore, the weights are now reset to $w_k^i = 1/N_s$. It is possible to implement this resampling procedure in operations by sampling N_s ordered uniforms using an algorithm based on order statistics." [6]. The resampling scheme algorithm is listed in 2.2

The intuitive idea of resampling is illustrated in Fig. 2.1. In the figure x_1 represents the state being approximated, the position of the circle (particle) represents the value of x_1 and

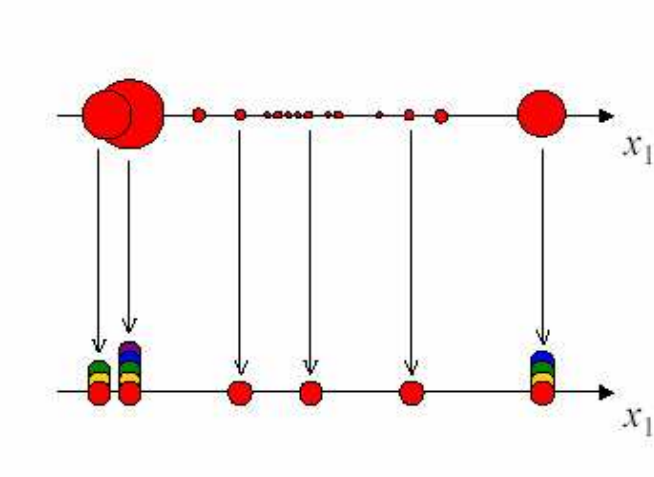


Figure 2.1: Idea of resampling

the radius of the circle represents the weight of the particle. The pdf of x_1 is represented by the particles and their corresponding weights. After resampling we see that all the particles have same weights (radius). What has happened in the process of resampling is that, all particles which had high weights were replicated in proportion to their weight. In this process particles with low weights are replicated a few times or completely lost. Resampling keeps the number of particles constant.

m	$w^{(m)}$	$r^{(m)}$
1	7/20	2
2	6/20	1
3	2/20	1
4	2/20	0
5	3/20	1

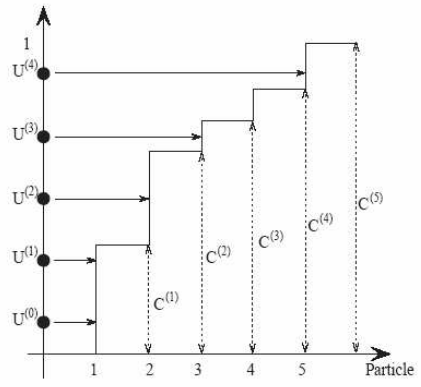


Figure 2.2: Resampling example

To understand the algorithm better an example is shown in Fig. 2.2. The table in the

Table 2.2: Resampling algorithm

$[\{x_k^{j*}, w_k^j, i^j\}_{j=1}^{N_s}] = RESAMPLE[\{x_k^i, w_k^i\}_{i=1}^{N_s}]$

1. Initialize the CDF: $c_1 = 0$
2. FOR $i = 2 : N_s$

   construct the CDF: $c_i = c_{i-1} + w_k^i$

   END FOR
3. Start at the bottom of the CDF: $i = 1$
4. Draw a starting point: $u_1 \sim \mathbf{U}[0, N_s^{-1}]$
5. FOR $j = 1 : N_s$

   Move along the CDF: $u_j = u_1 + N_s^{-1}(j-1)$

   WHILE $u_j \geqslant c_i$

   $i = i + 1$

   END WHILE

   Assign sample: $x_k^{j*} = x_k^i$

   Assign weight: $w_k^j = N_s^{-1}$

   Assign parent: $i^j = i$

   END FOR

figure lists the particle number m and its corresponding weight w^m. The plot alongside represents the CDF of the weights plotted vs particle number m. Now we draw samples $U^{(m)}$ uniformly between 0 and 1. Based on the samples we decide how many times to replicate a particle. We see how many samples $U^{(m)}$ fall within the contribution of a single particle and replicate that particle accordingly. For example U^0 and U^1 fall within the CDF contribution of particle 1. Therefore particle 1 is replicated 2 times as shown by the table in the column ($r^{(m)}$). On the other hand no samples fall within the contribution of particle 4, therefore it is not replicated as indicated in the table.

Though the resampling step reduces the effects of the degeneracy problem, it introduces a couple of other problems. Firstly, the opportunity to parallelize is limited since all the particles must be combined in the resampling step. Secondly, since the particles that have high weights are statistically selected many times it results in a loss of diversity among the particles as the resultant sample will contain many repeated points. This problem, which is known as sample impoverishment, is more severe in the cases of systems with small process noise. As a matter of fact, for the cases of very small process noise, all particles will collapse to a single point within a few iterations. "Thirdly, since the diversity of the paths of the particles is reduced, any smoothed estimates based on the particles' paths degenerate." [6]

The sequential importance sampling algorithm presented in section 2.5.1serves as the basis for most particle filters that have been developed so far. The various versions of particle filters in literature can be thought of as special cases of the general SIS algorithm. These special cases can be derived from the SIS algorithm by an approximate choice of importance sampling density or modifying the resampling step. We will consider two such particle filters proposed in literature. The particle filters considered are i) sampling importance resampling (SIR) filter; ii) auxiliary particle filter (APF)

2.5.3 Sequential importance resampling filter (SIR)

The SIR filter proposed in [5] is a Monte Carlo method that can be applied to bayesian filtering problems. The only requirements of the SIR filter are that the state dynamics $f_k(.,.)$ and $h_k(.,.)$ in 2.1 and 2.2 measurement functions are known, and we can draw samples from the process noise distribution v_{k-1} and the prior. Also the likelihood function $p(z_k|x_k)$ needs to be available for pointwise evaluation atleast upto proportionality.

The SIR algorithm can be derived from the SIS algorithm in the following way i)the importance density $q(x_k|x_{k-1}^i)$ is chosen to be the prior $p(x_k|x_{k-1}^i)$ and ii) the resampling step is applied at every time index.

The above choice of importance function requires that we need to draw samples from $p(x_k|x_{k-1}^i)$. A sample $x_k^i \sim p(x_k|x_{k-1}^i)$ can be generated by first generating a process noise sample $v_{k-1}^i \sim p_v(v_{k-1})$ and setting $x_k^i = f_k(x_{k-1}^i, v_{k-1}^i)$, where $p_v(.)$ is the known pdf of v_{k-1}. The choice of this importance density means the weights will be now given by

$$w_k^i \propto w_{k-1}^i p(z_k|x_k^i) \tag{2.45}$$

However since we resample at every time step, we have $w_{k-1}^i = 1/N \; \forall i$; therefore

$$w_k^i \propto p(z_k|x_k^i) \tag{2.46}$$

The weights given by 2.46 are normalized before the resampling each stage. An iteration of the algorithm is given below in Table 2.3

Though the SIR method is advantageous in the fact that the importance weights are easily evaluated and that the importance density can be easily sampled we must note that the importance sampling density for the SIR filter is independent of the measurement z_k, which means the state space is explored without any knowledge of the observations. Therefore the filter becomes very sensitive to the presence of outliers and is also inefficient. Also the process of resampling at every time instant can result in rapid loss of diversity of the

Table 2.3: SIR algorithm

$[\{x_k^{i*}, w_k^i\}_{i=1}^{N_s}] = SIR[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, z_k]$

1. FOR $i = 1 : N_s$

   Draw $x_k^i \sim p(x_k^i | x_{k-1}^i)$

   Calculate $w_k^i = p(z_k | x_{k-1}^i)$

   END FOR

2. Calculate total weight: $t = \sum_{i=1}^{N_s} w_k^i$

3. FOR $i = 1 : N_s$

   Normalize: $w_k^i = t^{-1} w_k^i$

   END FOR

4. Resample using $[\{x_k^i, w_k^i, -\}_{i=1}^{N_s}] = RESAMPLE[\{x_k^i, w_k^i\}_{i=1}^{N_s}]$

particles.

2.5.4 Auxiliary Particle Filter (APF)

The auxiliary particle filter was introduced by Pitt and Shephard [12] as a variation of the standard SIR filter. This filter can be obtained from the SIS framework by modifying the importance density $q(x_k | x_{k-1}^i, z_k)$ to be $q(x_k, i | z_{1:k})$,which samples the pair $\{x_k^j, i^j\}_{j=1}^{N_s}$ where i^j refers to the index f the particle at time step $k-1$.

By Bayes' rule we can obtain $p(x_k, i | z_{1:k})$

$$\begin{aligned} p(x_k, i | z_{1:k}) &\propto p(z_k | x_k) p(x_k, i | z_{1:k-1}) \\ &= p(z_k | x_k) p(x_k | i, z_{1:k-1}) p(i | z_{1:k-1}) \\ &= p(z_k | x_k) p(x_k | x_{k-1}^i) w_{k-1}^i \end{aligned} \tag{2.47}$$

The APF filter operates by sampling from the joint density $p(x_k, i | z_{1:k})$ and then omits the indices i in the pair (x_k, i) to produce a sample $\{x_k^j\}_{j=1}^{N_s}$ from the marginalized density

$p(x_k|z_{1:k})$. The importance density used to draw samples $\{x_k^j\}_{j=1}^{N_s}$ is defined to satisfy the proportionality

$$q(x_k, i|z_{1:k}) \propto p(z_k|\mu_k^i)p(x_k|x_{k-1}^i)w_{k-1}^i \tag{2.48}$$

where μ_k^i is some characterization of x_k given x_{k-1}. This could be the mean, in which case $\mu_k^i = E[x_k|x_{k-1}^i]$ or as in our case a sample $\mu_k^i \sim p(x_k|x_{k-1}^i)$

We can also show that the weights at every time step will be assigned according to

$$w_k^j \propto w_{k-1}^{i^j} \frac{p(z_k|x_k^j)p(x)k^j|x_{k-1}^{i^j})}{q(x_k^j, i^j|z_{1:k})} = \frac{p(z_k|x_k^j)}{p(z_k|\mu_k^{i^j})} \tag{2.49}$$

The algorithm for the APF is given in Table. 2.4. Compared with the SIR filter the advantage of the APF filter is that it generates points from the sample at $k-1$ which, conditioned on the current measurement, are most likely to be close to the true state of the target. The APF can be thought of a resampling at a previous step, based on some point estimate μ_k^i that represents $p(x_k|x_{k-1}^i)$. The APF's performance is very much affected by the process noise of the system. If the process noise is too strong then it may happen that μ_k^i does not characterize $p(x_k|x_{k-1}^i)$ well enough and the result is a poor set of particles. In such cases the use of APF degrades the performance of the tracking system.

Table 2.4: APF algorithm

$[\{x_k^{i*}, w_k^i\}_{i=1}^{N_s}] = APF[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, z_k]$

1. FOR $i = 1 : N_s$

   Calculate μ_k^i

   Calculate $w_k^i = q(i|z_{1:k}) \propto p(z_k \mu_k^i) w_{k-1}^i$

   END FOR

2. Calculate total weight: $t = \sum_{i=1}^{N_s} w_k^i$

3. FOR $i = 1 : N_s$

   Normalize: $w_k^i = t^{-1} w_k^i$

   END FOR

4. Resample using $[\{-, -, i^j\}_{j=1}^{N_s}] = RESAMPLE[\{x_k^i, w_k^i\}_{i=1}^{N_s}]$

5. FOR $j = 1 : N_s$

   Draw $x_k^i \sim p(x_k^i | x_{k-1}^{ij})$

   Assign $w_k^j = \frac{p(z_k|x_k^j)}{p(z_k|\mu_k^{ij})}$ END FOR

6. Calculate total weight: $t = \sum_{i=1}^{N_s} w_k^i$

7. FOR $i = 1 : N_s$

   Normalize: $w_k^i = t^{-1} w_k^i$

   END FOR

CHAPTER 3

Target tracking using particle filters

Our aim will be to detect the 2-D spatial location of the target and estimate its current aspect. For simplicity let us consider the case of a single target being present in the scene. The following sections develop the target motion, target signature, and the background clutter models that form the basis of our detection and tracking algorithms.

3.1 System models

3.1.1 Target motion model

We assume the data we observe is sampled from a set of continuous valued state variables at a constant sampling rate determined by the frame rate of the camera. Let k be a non-negative integer number and denote by Δ the time interval between two consecutive measurements. The state vector at instant $t = k\Delta$ of a target typically consists of the position, x_k and y_k, and the velocity, $\dot{x_k}$ and $\dot{y_k}$, of the target centroid in a 2D Cartesian coordinates system: $\mathbf{x}_k = [x_k\ \dot{x_k}\ y_k\ \dot{y_k}]$. The random sequences of the centroid position and the velocity are assumed to be statistically independent and evolve in each dimension in time according to the white noise acceleration model [3].

The state update equation can be written as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{3.1}$$

where the transitional matrix $\mathbf{F} = [1\ \Delta;\ 0\ 1]$, and the process noise $\mathbf{w}_k$ is assumed to be white, zero-mean Gaussian noise [13].

3.1.2 Target signature model

We assume that in any frame, the clutter free target image is contained within a rectangular region of size $(r_i + r_s + 1)$x$(l_i + l_s + 1)$.Here r_i and r_s denote the maximum vertical pixel distances in the target template image when we move up and down respectively from the target centroid. Similarly l_i and l_s are the maximum horizontal pixel distances in the target image when we move away, respectively left and right, from the centroid of the target.

For each centroid position $(i, j)\epsilon\mathcal{L}$ where $\mathcal{L} = (i, j)|1 \leq i \leq L, 1 \leq j \leq M$ the nonlinear function H returns a spatial distribution of real valued pixel intensities $a_{k,l}, -r_i \leq k \leq r_s, -l_i \leq l \leq l_s$, centered at (i,j). We can write

$$H(i_n, j_n) = \sum_{k=-r_i}^{r_s} \sum_{l=-l_i}^{l_s} a_{k,l} E_{i_n+k, j_n+l} \tag{3.2}$$

where $E_{r,s}$ is an L x M matrix whose entries are all zeros except for the element (r,s) which is equal to 1.The coefficients $a_{k,l}$ in (3.2) are referred to as the target signature parameters.

3.2 Clutter model

The clutter frames $\{\mathbf{v}_k\}$ is assumed to be an independent, identically distributed (i.i.d.) Gaussian random sequences relative to the processing noise $\mathbf{w}_k$ in the motion model, also it is statistically independent of $\mathbf{x}_k$, and with zero mean and non-singular covariance matrices respectively. It is described by the first order, non-causal Gaussian Markov random field (GMRF) model in each frame [14]:

$$\mathbf{v}_n(i, j) = \beta_v^c[v_n(i-1, j) + v_n(i+1, j)] + \beta_h^c[v_n(i, j-1) + v_n(i, j+1)] + \varepsilon_n(i, j)\ . \tag{3.3}$$

where $E[v_n(i, j)\varepsilon_n(p, r)] = \sigma_c^2\delta_{i-p,j-r}$ and the unknown parameters β_v^c and β_h^c are, respectively, the vertical and horizontal predictor coefficients, and ε_k is the prediction error [15]. We also assume that the clutter frames v_n are statistically independent. Sample clutter frames generated with different parameters is shown in Fig. 3.1

3.3 Observation model

An infrared sensor device sequentially generates raw measurements of a surveillance region that contain both target of interest and spurious heat sources (clutter). The raw measurements at instant k are sampled and processed to form a 2D digital sensor image called a frame. Each frame k is modelled using a L x M matrix, $\mathbf{Y}_k$ which collects the measured 2D frames. Fig. 3.4 shows how each observation is generated according to 3.4

$$\mathbf{Y}_k = \mathbf{H}(\mathbf{x}_k) + \mathbf{v}_k \ . \tag{3.4}$$

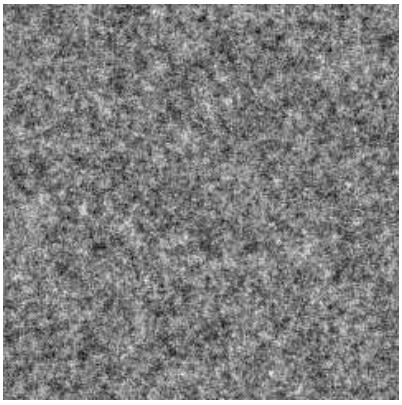
3.4 Algorithm implementation

The first step in the particle filter algorithm is to initialize the particles or in other words draw samples from the prior. In initialization, the particles are drawn from some known prior $p\left(\mathbf{x}_0\right)$. Instead of picking particles from a uniformly distributed prior which conveys no specific information about the location of the target, we draw particles according to a *possibility map* $\mathbf{M}_k(i,j)$, which is obtained by convolving the initial observation $\mathbf{Y}_0(i,j)$ first with the template of the GMRF noise model $\mathbf{H} = [0 \ -\beta_v \ 0; \ -\beta_h \ 1 \ -\beta_h; \ 0 \ -\beta_v \ 0]$, and then the target signature template model $\mathbf{G}_0(i,j)$. The parameters β_h and β_v are estimated using the approximate maximum likelihood (AML) algorithm [14].

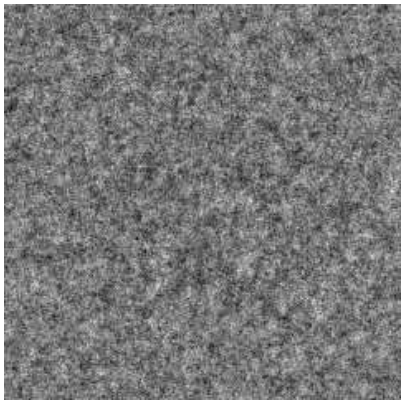
$$\mathbf{M}_0(i,j) = \mathbf{G}_0(i,j) * [\mathbf{Y}_0(i,j) * \mathbf{H}] \ . \tag{3.5}$$

The figures corresponding to 3.5 are shown in Fig. 3.3, the bright points in image (c) correspond to possible positions of the target. The final initialization of the particles is shown in Fig. 3.2. The dark circles indicate the position of the particles, to make the illustration clear only a 100 particles are used in this example. Notice how a number of particles are clustered around the actual position of the target indicated by * in the figures, this indicates a good initialization of the particles.

Likelihood computation is a critical step in particle filtering as the weights are assigned in proportion to the likelihood value of each particle. A proper likelihood function will guide particles towards the true position of the target, as it directly determines the weight of each particle. Actually, it is a measurement of how exactly a particle is consistent with the current observation or in other words it is a measure of how likely is it that the particle under consideration produced the current observation. Following this point, we use a similar function as the one we used in the initialization: $p\left(\mathbf{Y}_k|\mathbf{x}_k\right) \propto \mathbf{M}_k(i,j)$. It is the similarity between the observation and the template formed by the state variables of the particle under consideration. The only difference is, instead of computing the whole similarity map, we only find the possibility value at one position for a given particle.

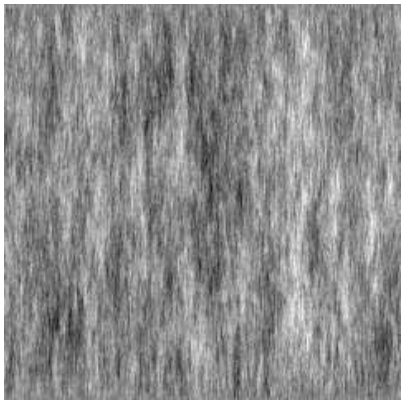


$\beta_h^c = 0.24, \beta_v^c = 0.25, \sigma_c^2 = 25$

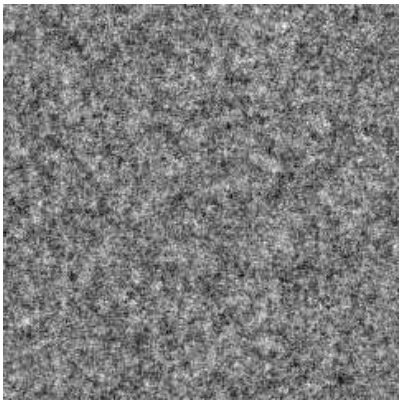


$\beta_h^c = 0.24, \beta_v^c = 0.25, \sigma_c^2 = 64$

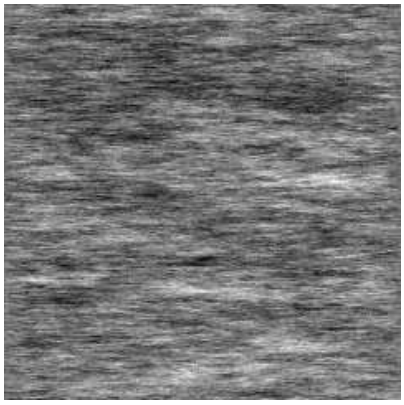
$\beta_h^c = 0.49, \beta_v^c = 0.01, \sigma_c^2 = 25$



$\beta_h^c = 0.01, \beta_v^c = 0.49, \sigma_c^2 = 25$



$\beta_h^c = 0.24, \beta_v^c = 0.24, \sigma_c^2 = 128$



$\beta_h^c = 0.49, \beta_v^c = 0.01, \sigma_c^2 = 128$

Figure 3.1: Sample clutter frames with different parameters

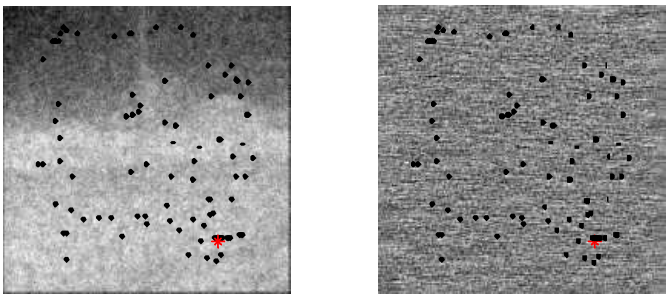
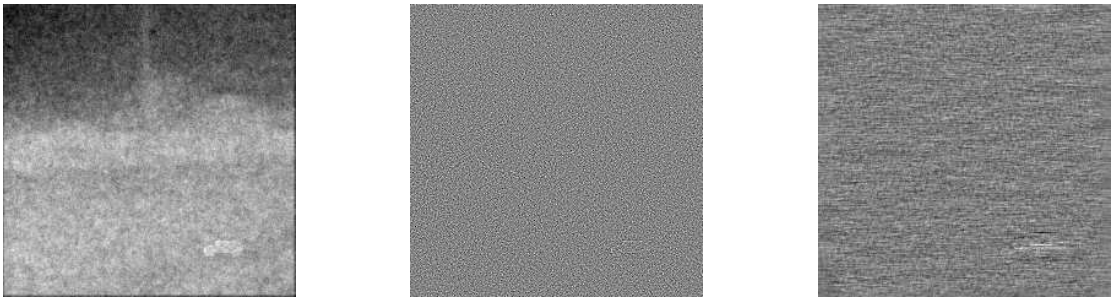


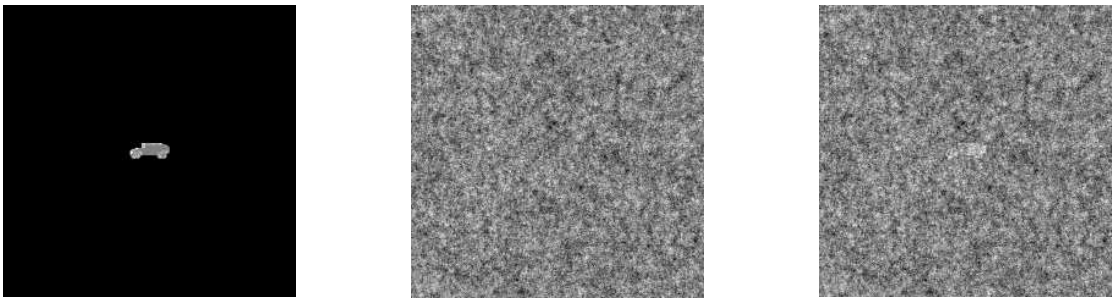
Figure 3.2: Initialization of particles



(a) (b) (c)

$\mathbf{Y}_0(i,j)$ $\mathbf{Y}_0(i,j) * \mathbf{H}$ $\mathbf{G}_0(i,j) * [\mathbf{Y}_0(i,j) * \mathbf{H}]$

Figure 3.3: Initialization steps



$\mathbf{H}(\mathbf{x}_k)$ $\mathbf{v}_k$ $\mathbf{Y}_k$

Figure 3.4: Observation model

CHAPTER 4

Rotation and scaling invariant target tracking using particle filters

Though a lot of previous work address the problem of target tracking only a few have tried to answer the question of multi-aspect target tracking. In our work we have tried to address this problem by incorporating an affine model to account for random changes in scale and rotation of the target signature as a first step towards multi aspect target tracking. The details of our affine model and the results of our simulations are discussed in this chapter.[1].

4.1 Affine transformation

An affine transformation is any transformation that preserves collinearity and ratios of distances. In general, it is a composition of rotations, translations, scaling, and shears of the basic template. A shear preserving horizontal lines has the form $(x, y) \longrightarrow (x + \alpha y, y)$, where α is the shearing factor. To model the different aspects of the target in motion, we apply the following affine transformation to the base template in each frame:

$$\mathbf{T} = \begin{bmatrix} 1 & \alpha & -\alpha y \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & (1-s_x)x \\ 0 & s_y & (1-s_y)y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & (1-\cos\theta)x - (\sin\theta)y \\ -\sin\theta & \cos\theta & (1-\cos\theta)y + (\sin\theta)x \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where α, s_x, s_y and θ are the shearing, scaling along x-axis, y-axis and rotation parameter, respectively. All of them are continuous-valued random variables which follow the first order Markov chain.

[1] Joint work with Dr Tang Li

Process noise (γ_α, γ_{s_x}, γ_{s_y}, γ_θ) is added to each random variable to reduce the quantization effect on parameter estimation. Therefore, we define a new augmented state vector as:

$$\mathbf{x}_k = [x_k \; \dot{x}_k \; y_k \; \dot{y}_k \; s_x \; s_y \; \alpha \; \theta] \; . \tag{4.2}$$

4.2 Rotation and scaling model

The rotation model is defined to be

$$\theta_{k+1} = \mathbf{H}(\theta_\mathbf{k}) + \gamma_\theta \tag{4.3}$$

in which the function H is is a first order Markov chain with equal transition probability (i.e., 1/3) of increasing, decreasing by a quantization step (Δ_θ) or staying at the same value in each time instant. The rotation angles vary within $[-30°, 30°]$ degrees with a step size $\Delta_\theta = 2°$. The function $\mathbf{H}$ is shown in Fig. 4.1. To make the rotation state variable a continuous valued variable a small process noise γ_θ is added to the rotation angle. This noise is uniformly distributed with zero mean and its variance depends on the step size Δ_θ.

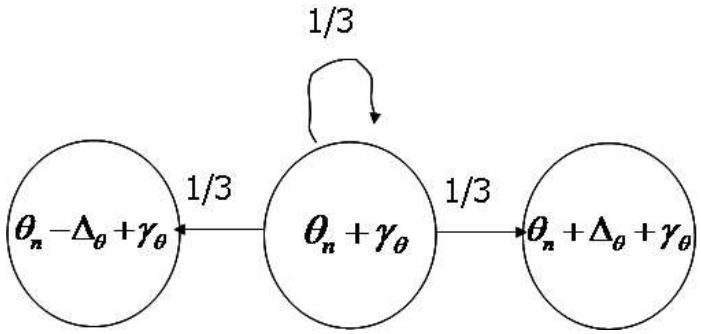


Figure 4.1: First order Markov model for affine parameters

The target scale vector is also modelled as a first order Markov chain with equal transition probability (i.e., 1/3) of increasing, decreasing by a quantization step (Δ_{s_x}) or staying at

the same value in each time instant. The scaling factors vary within $[0.5, 1.5]$ with a step size of $\Delta_{s_x} = \Delta_{s_x} = 0.05$. We also add to it a uniform zero mean process noise depending on the step size.

Due to the small size of the target template we do not consider shearing effects on the target. But the shearing variable may be modelled similar to the rotation or scale variables and can be estimated using the same particle filtering methods.

4.3 Multi aspect tracking using SIR and APF

The tracking algorithm is implemented in a similar way as in the single aspect tracking case, but now the state vector also include the added affine parameters to account for random scaling and rotation of the target template.

The initial template of the target $\mathbf{G}_0(i, j)$ is obtained by applying the mean affine parameters to the base template (i.e. $s_x = s_y = 1, \theta = 0$ and $\alpha = 0$). Since with the affine model, we add additional variables to the state vector, we hope the true position of the target will be included in the initial set of the particles. The pseudo-code algorithm for multi aspect target tracking using SIR and APF is given in Table. 4.1

4.4 Simulation and results

For all our simulations we used the target template shown in Fig. 4.2. The size of the original template is 15x35 pixels. For the purpose of simulation the original template was attenuated by a factor of 0.05. The mean intensity of the template after attenuation is 9.9509 and it has an energy of 179.721. The variance of the GMRF noise is 25, resulting in a SNR of 8.566dB. Since the target is very small, we set the shearing factor to zero to avoid significant distortion.

To generate the ground truth data we initialed the target's initial centroid position to be randomly distributed uniformly between pixels 160 and 200 in the x direction and between

Table 4.1: Pseudo-code of the algorithms

Implementation of SIR Filter	Implementation of APF Filter
1. Initialization	1. Initialization: same as SIR
$for j = 1, \cdots, N_p$	2. for j=1,$\cdots, N_p$
Draw $x_{0,1}^j \sim p(x_{0,1}), x_{0,2}^j \sim p(x_{0,2})$	Draw $\widetilde{\mu}_{n,1}^j \sim p(x_{n,1}\|x_{n-1,1}^j)$
$r_0^j \sim p(r_0),\ \ s_{0,x}^j \sim p(s_{0,x}), s_{0,y}^j \sim p(s_{0,y})$	$\widetilde{\mu}_{n,2}^j \sim p(x_{n,2}\|x_{n-1,2}^j), \widetilde{\mu}_{n,r}^j \sim p(r_n\|r_{n-1}^j)$
and set $w_0^j = 1/N_p, n = 1$	$\widetilde{\mu}_{n,sx}^j \sim p(s_{n,x}\|s_{n-1,x}^j)$,
end	$\widetilde{\mu}_{n,sy}^j \sim p(s_{n,y}\|s_{n-1,y}^j)$ and compute
2. for j=1,$\cdots, N_p$	$\widehat{w}_n^j \propto p(y_n\|\widetilde{\mu}_{n,1}^j, \widetilde{\mu}_{n,2}^j, \widetilde{\mu}_{n,r}^j, \widetilde{\mu}_{n,sx}^j, \widetilde{\mu}_{n,sy}^j)$
Draw $\widetilde{x}_{n,1}^j \sim p(x_{n,1}\|x_{n-1,1}^j)$	end
$\widetilde{x}_{n,2}^j \sim p(x_{n,2}\|x_{n-1,2}^j), \widetilde{r}_n^j \sim p(r_n\|r_{n-1}^j),$	3. Normalize such that $\sum_{j=1}^{N_p} \widehat{w}_n^j = 1$
$\widetilde{s}_{n,x}^j \sim p(s_{n,x}\|s_{n-1,x}^j)$,	4. for $j = 1, \cdots, N_p$
$\widetilde{s}_{n,y}^j \sim p(s_{n,y}\|s_{n-1,y}^j)$ and compute	Draw $k^j \sim \{1, 2, ..., N_p\}$ such that
$w_n^j \propto p(y_n\|\widetilde{x}_{n,1}^j, \widetilde{x}_{n,2}^j, \widetilde{r}_n^j, \widetilde{s}_{n,x}^j, \widetilde{s}_{n,y}^j)$	$p(k^j = i) = \widehat{w}_n^i$ for $i = 1, 2, ..., N_p$
end	Draw $x_{n,1}^j \sim p(x_{n,1}\|x_{n-1,1}^{k^j})$,
3. Normalize such that $\sum_{j=1}^{N_p} w_n^j = 1$	$x_{n,2}^j \sim p(x_{n,2}\|x_{n-1,2}^{k^j}), r_n^j \sim p(r_n\|r_{n-1}^{k^j}),$
4. Resample to generate a new set of	$s_{n,x}^j \sim p(s_{n,x}\|s_{n-1,x}^{k^j})$,
samples $[x_{n,1}^j, x_{n,2}^j, r_n^j, s_{n,x}^j, s_{n,y}^j]$ such	$s_{n,y}^j \sim p(s_{n,y}\|s_{n-1,y}^{k^j})$ and compute
that $p([x_{n,1}^j, x_{n,2}^j, r_n^j, s_{n,x}^j, s_{n,y}^j] =$	$w_n^j \propto \frac{p(y_n\|x_{n,1}^j, x_{n,2}^j, r_n^j, s_{n,x}^j, s_{n,y}^j)}{p(y_n\|\widetilde{\mu}_{n,1}^{k^j}, \widetilde{\mu}_{n,2}^{k^j}, \widetilde{\mu}_{n,r}^{k^j}, \widetilde{\mu}_{n,sx}^{k^j}, \widetilde{\mu}_{n,sy}^{k^j})}$
$[\widetilde{x}_{n,1}^k, \widetilde{x}_{n,1}^k, \widetilde{r}_n^k, \widetilde{s}_{n,x}^k, \widetilde{s}_{n,y}^k]) = w_n^k$	end
5. Set $w_n^j = 1/N_p$ for $1 \leq j \leq N_p$ and	5. Normalize such that $\sum_{j=1}^{N_p} w_n^j = 1$
estimate the mean values	6. Estimate the mean values
6. Set $n = n + 1$ and return to step 2	7. Set $n = n + 1$ and return to step 2

Figure 4.2: Target template

pixels 140 and 180 in the y direction. The initial velocities along both directions were initialized to be normal random variables of zero mean and unit variance. In addition to the white noise acceleration model a constant drift of -2 and -0.5 pixels/frame were added in the x and y directions respectively.

The sensor observation image is of size 239 x 239 pixels. We generated the background image from a real infrared video by subtracting from the real infrared image its spatially variant local mean and then fitting a first order GMRF field to the resultant image. The parameters of the GMRF field are estimated using the Approximate Maximum Likelihood (AML) algorithm discussed in [14]. Then using these parameters we generate random GMRF field to which we add the local means from the original image to create a cluttered background. To this cluttered background we add the target template to generate the final observation. The original IR image and the cluttered image are shown in figure 4.3

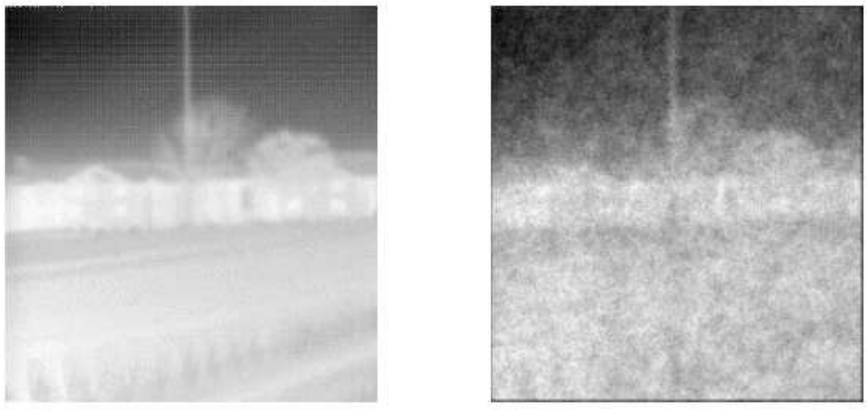


Figure 4.3: (a) Original Infrared image (b) Simulated cluttered background

We assumed the time interval between two consecutive measurements to be $\Delta = .075$. We tried to track a target moving according to the white noise acceleration model and with rotation and scaling according to the model in 4.3 over 50 time frames. Three sample observations are shown below in Fig. 4.4

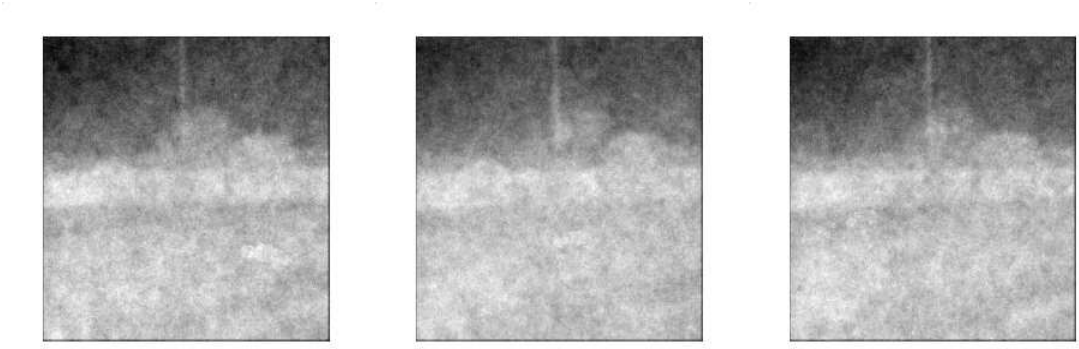


Figure 4.4: Sample observation 10'th, 35'th frames and 50'th frames

Each of the particle filter runs were simulated with 3500 particles. The particles were either initialized according to the proposed initialization method or were uniformly distributed around the true position of the target. The effects of the initialization is discussed in the following sections.

4.4.1 Importance of affine parameters

The infrared signature of most real world targets keep varying over time resulting in different target signatures on the sensor. To model these varied signatures we have included in our state vector the rotation, scaling and shearing variables. To demonstrate the effect of including these affine parameters, we tried to run the particle filtering algorithms (SIR and APF) without considering the affine parameters. The results are shown in Fig. 4.5

As seen from Fig. 4.5 when we do not consider the affine parameters both filters perform poorly especially the SIR filter which completely looses track of the target and the estimated trajectory begins to deviate from the ground truth trajectory. When affine parameters are considered the mean position error is quite small, usually within 1 to 2 pixels,

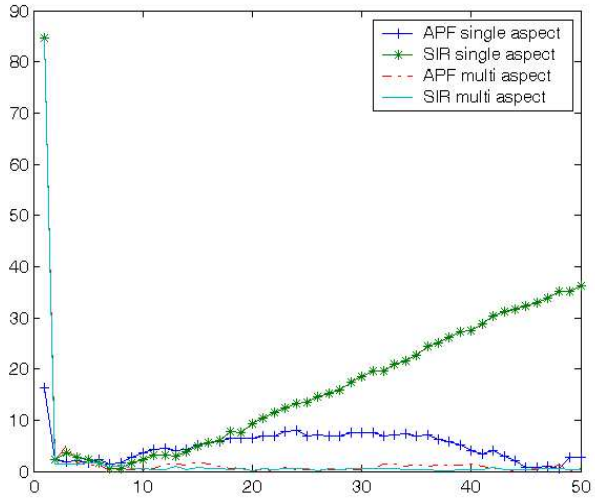


Figure 4.5: Comparison of mean tracking error with and without affine parameters

which is quite good. Thereby we have shown that considering affine parameters improves the tracking ability of the particle filter.

4.4.2 Comparison between SIR and APF

In this section we compare and evaluate the tracking performance of the SIR and APF implementations. We also study the contribution of the match filtering based initialization. There are four criteria to be used for algorithm evaluation, i.e., the number of convergent runs out of 15 Monte Carlo runs, the mean convergence time (the first time when the tracking error of position is below 1 pixel), the mean and the standard deviation of estimation errors for the four state variables (x position,y position,rotation and scale). The results of 15 Monte Carlo runs for SIR and APF filters, with and without initialization, are shown in Table 4.2.

It is seen from the table that both SIR and APF can effectively estimate all state variables, including positions, rotation and scaling parameters. This demonstrates the capability and robustness of particle filters to deal with tracking problems with strong nonlinearity

Table 4.2: Comparison of SIR and APF performance with affine parameters

Algorithm	Initialization (INIT)	Steps to converge	Mean error in			Std Dev of error in		
			position (x, y)	rotation θ	scale s	position (x, y)	rotation θ	scale s
APF	with INIT(12)	7.75	0.8652	1.1322	0.0212	0.8227	2.1387	0.0358
	w/o INIT(10)	10.90	1.1888	1.5535	0.0379	1.1927	2.8408	0.0567
SIR	with INIT(12)	9.42	1.0490	1.1551	0.0342	0.7423	1.6371	0.0544
	w/o INIT(4)	13.75	1.5666	2.0995	0.0636	1.6337	3.9023	0.0865

(N) indicates how many monte carlo runs out of 15 converged

in the target motion model. Specifically, the APF filter outperforms the SIR filter almost in all aspects, showing the usefulness of the two-step sampling process with the consideration of present observations. Also, the matching filtering based particle initialization is helpful to improve the number of convergent results and also the effectiveness of state estimation, in both the SIR and APF case.

The tracking results at a number of time steps for the APF and SIR filter with initialization are shown in Fig. 4.6, we notice that the particles converge to the true position of the target within the first few time steps itself. Then all of the particles keep tracking the target even though we can hardly see it with our eyes in the original observation sequence. The first column and the third column show the particle distribution at the k'th time step for the APF and SIR filter respectively. The second and the fourth column show the actual position of the car and the bounding box represents the estimated position. The size and angle of orientation of the box represents the aspect (scale factor and rotation angle) estimate. A bounding box which fits exactly around the target represents an accurate estimate of both position and the aspect of the target.

In particular at $k = 1$ in Fig. 4.6 the particles are distritbuted according to the possibility map. Particles are produced where they have the most significant values in the possibility

map, which turns out to always contain the true position of the target both in the APF and SIR case. Then at time step $k = 2$ almost all of the particles converge to the actual position of the target and as time progresses the particles converge to the target centroid and begin to track its position along with the aspect. By contrast in the case without initialization in Fig. 4.7, where the particles are uniformly distributed at first, in the time step $k = 2$ we see that not all the particles have converged to the actual position of the target, thereby proving that initialization helps the particles to converge faster to the actual position of the target. The plots of the position state variables and the corresponding errors are shown in Fig. 4.8, Fig. 4.9, Fig. 4.10 repectively, the results for the rotation angle estimation are shown in Fig. 4.11 and the corresponding error is shown in Fig. 4.12. Similar estimation results for the scaling factor ar shown in Fig. 4.13 and the corresponding error is shown in Fig. 4.14. Discussions about the results are alongside the figure itself.

Based on the results discussed till now we can conclude that the APF filter is a better algorithm than the SIR filter and the initialization step helps the algorithm to converge to the true state faster.

4.4.3 Influence of noise

From the above section we see that the APF filter with initialization provides the best tracking results, so in this section we will examine the effect of noise on the performance of the APF filter with initialization. To demonstrate the robustness of the algorithm to noise, we run it at different noise levels independently with the energy of the target unchanged. The noise level is changed by increasing the variance of the GMRF clutter field. The results of the experiment are shown in Table 4.3.

As the noise level keeps increasing the algorithm manages to converge to the true position of the target but take a longer time to converge. Also we notice the mean error also increases as the noise power increases. Finally when the noise variance was 36 the estimated position failed to converge within 1 pixel of the actual position within 50 time steps.

Table 4.3: Tracking performance under different noise levels

Variance of the noise	4.1383	9	25	36
Corresponding SNR (dB)	10.055	6.681	2.244	0.660
Mean tracking error	0.8652	1.4629	2.3544	9.0098
Steps to converge($\leq$1 pixel)	7.75	8.00	13.00	-

Thereby we can conclude that the particle filtering technique is quite effective in estimating the target trajectory under heavy noise conditions even when the target is almost invisible to the naked eye.

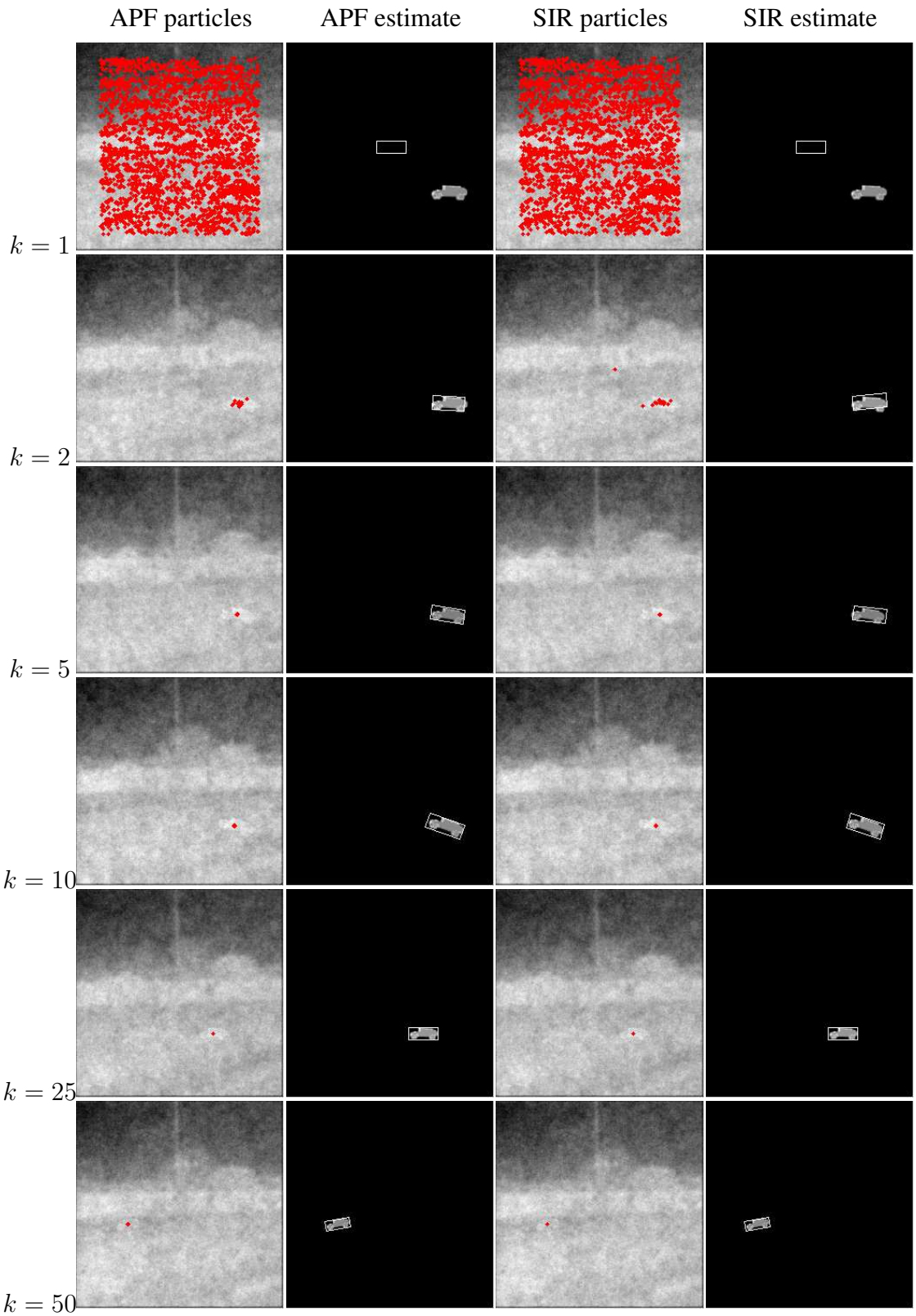


Figure 4.6: Tracking results at time steps 1,2,5,10,25 and 50 for SIR and APF with initialization

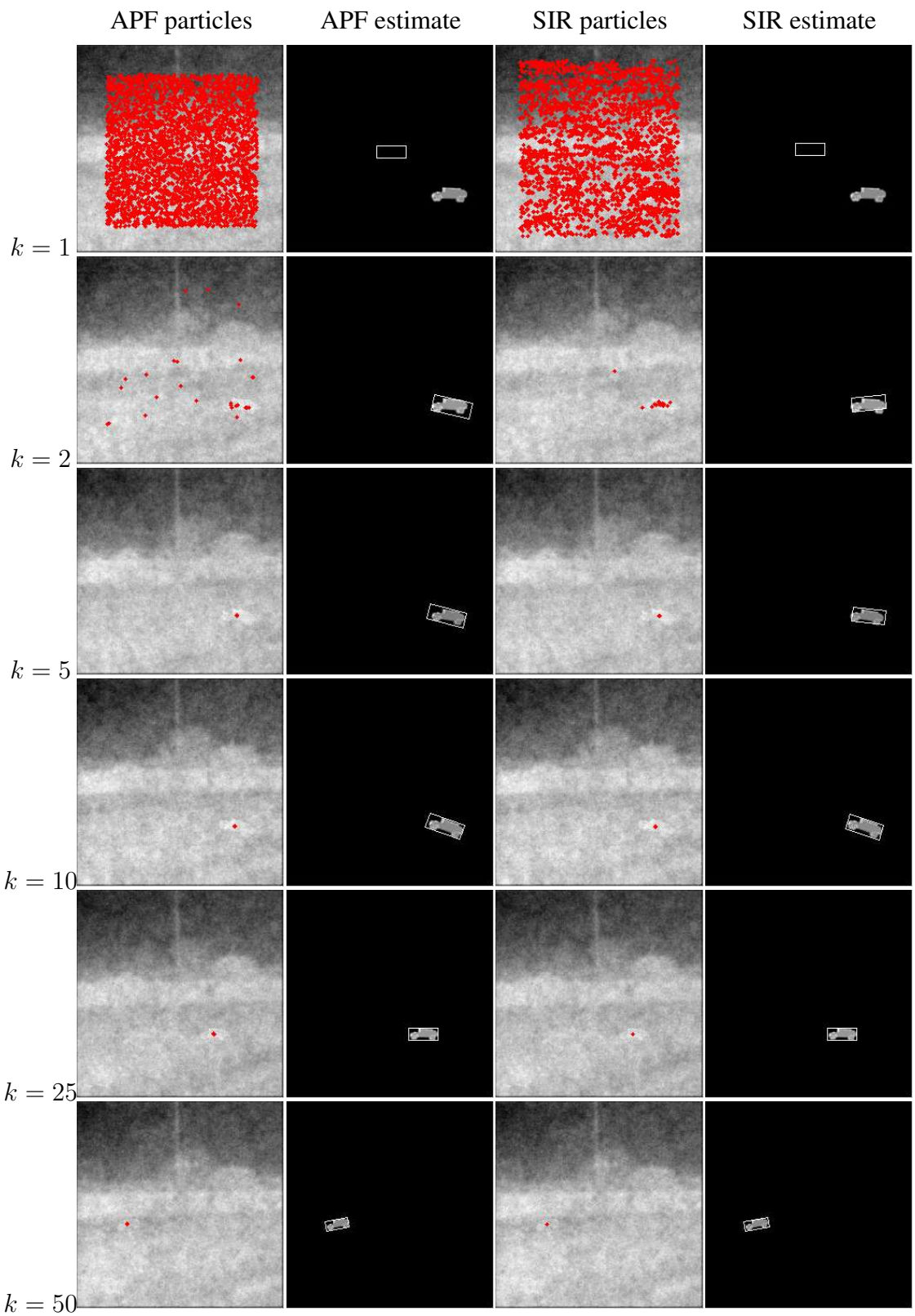
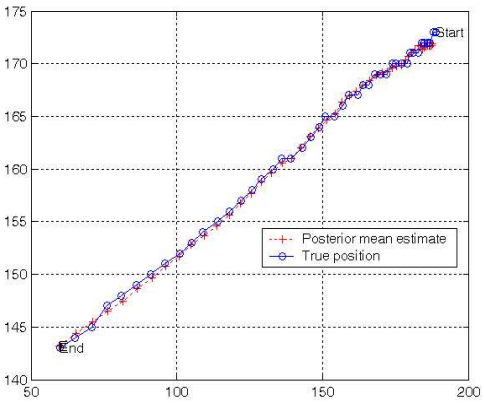
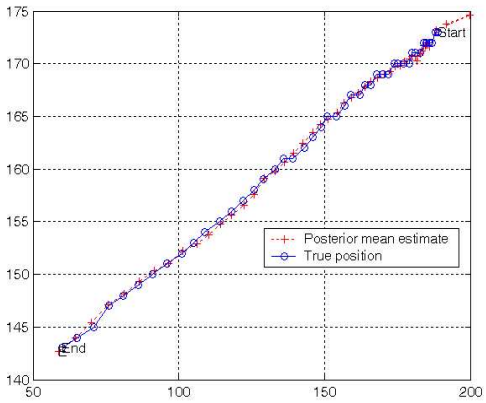


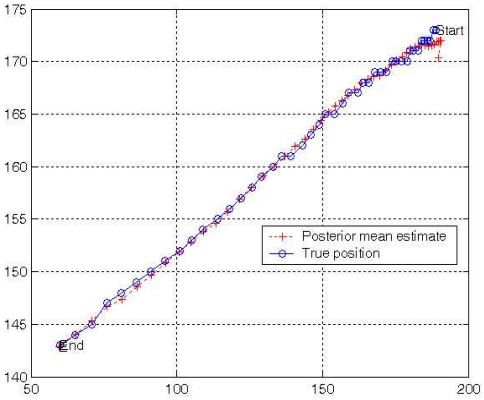
Figure 4.7: Tracking results at time steps 1,2,5,10,25 and 50 for SIR and APF without initialization



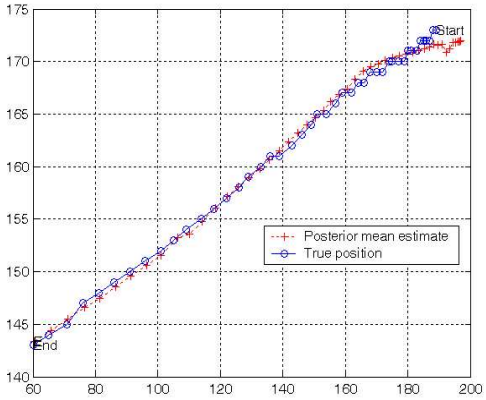
APF with initialization



APF without initialization

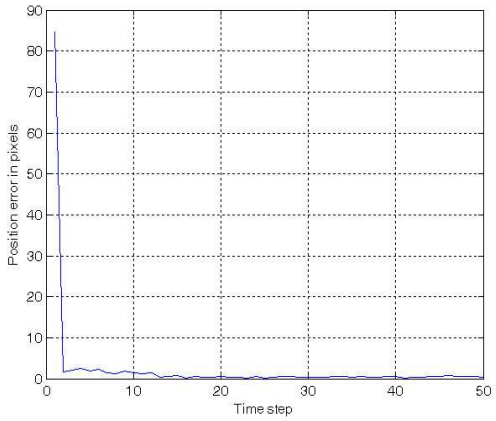


SIR with initialization

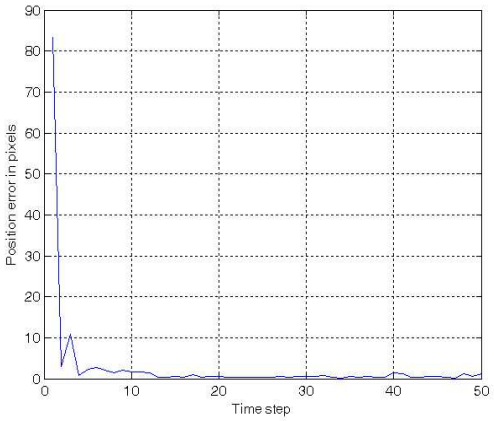


SIR without initialization

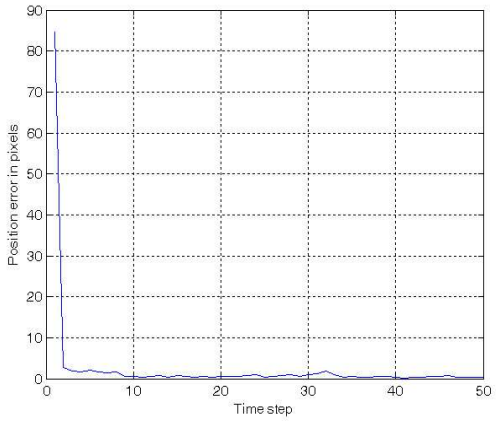
Figure 4.8: shows the actual and the estimated trajectory of the target over 50 time steps for the APF and SIR filter with and without initialization. From these figures we are not able to decide any major differences except for the fact that all the methods are effectively able to track the target.



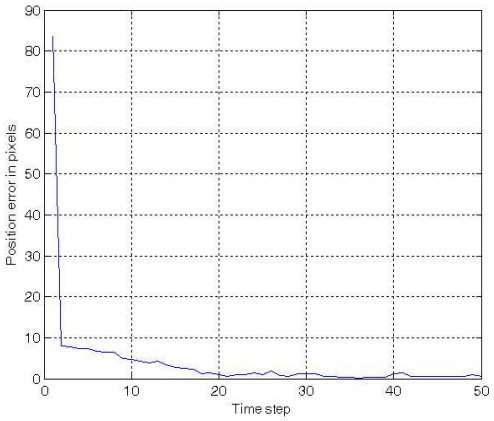
APF with initialization



APF without initialization

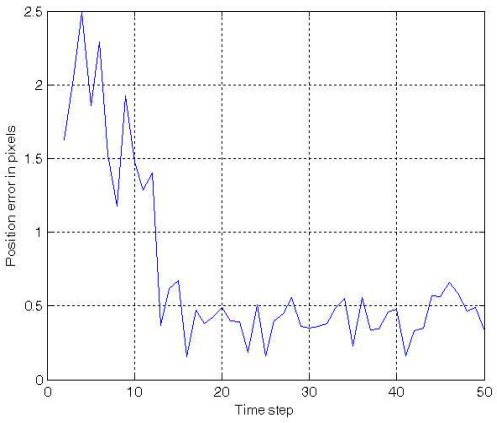


SIR with initialization

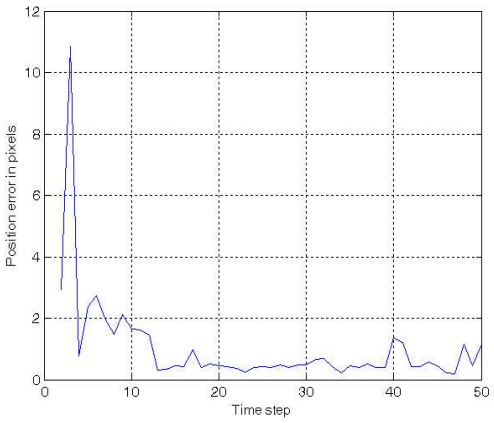


SIR without initialization

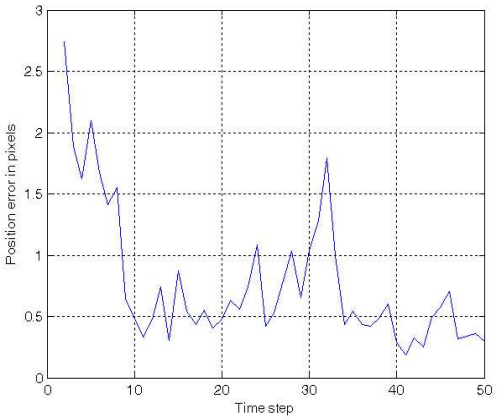
Figure 4.9: shows the euclidian error in the position estimate of the object from the first time step. At time step $k = 1$ the error is large because the particles all have equal weights in the first time step and thereby the mean estimate is quite far from the actual position of the target. From Fig. 4.9 we can see overall that the methods with initialization converge faster to the actual position than those without initialization.



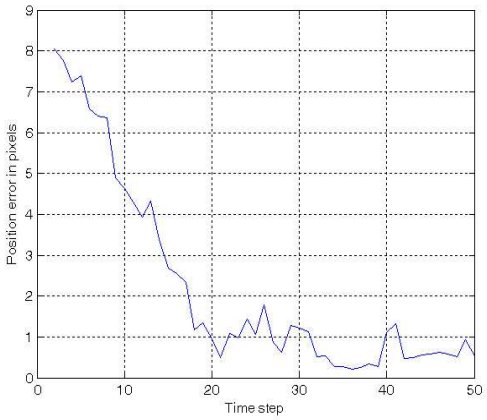
APF with initialization



APF without initialization

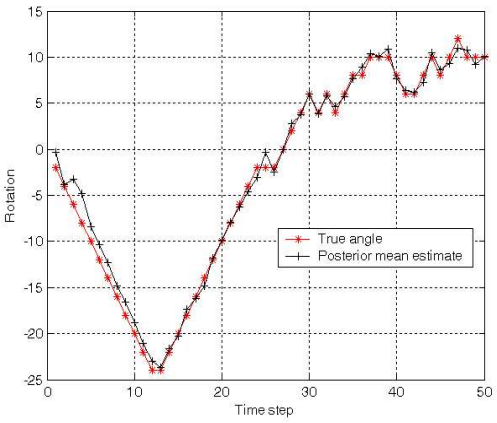


SIR with initialization

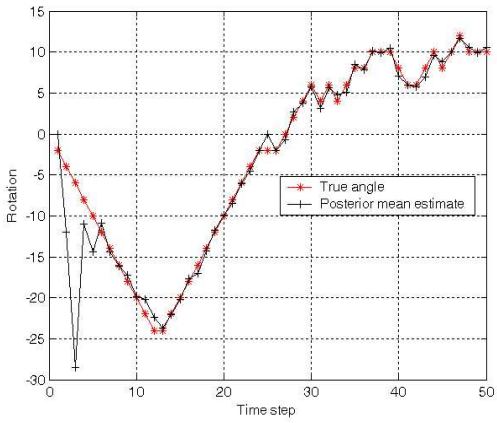


SIR without initialization

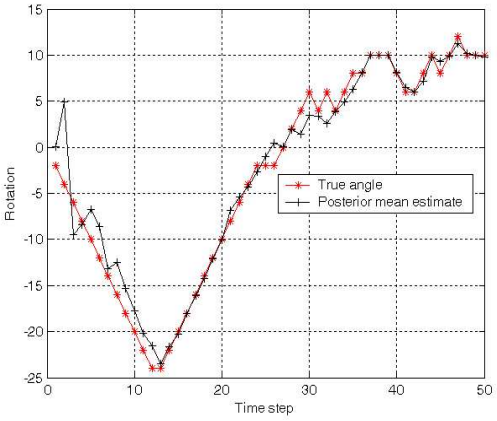
Figure 4.10: shows the euclidian error in the position but this time from the 2'nd time step $(k = 2)$. This plot will help us to compare the APF and SIR performances more closely. We see that the APF with initialization shows the best performance by converging fast and the error remains below the 0.5 pixel level majority of the time. The SIR filter with initialization also converges fast but the error oscillates between 0.5 pixel and 1 pixel level. The APF without initialization converges slowly compared to the previous two methods and the error is between 1 and 2 pixels most of the time. The SIR filter without initialization takes the longest time to converge to the true position and the error is significantly above 1 pixel for most of the tracking period.



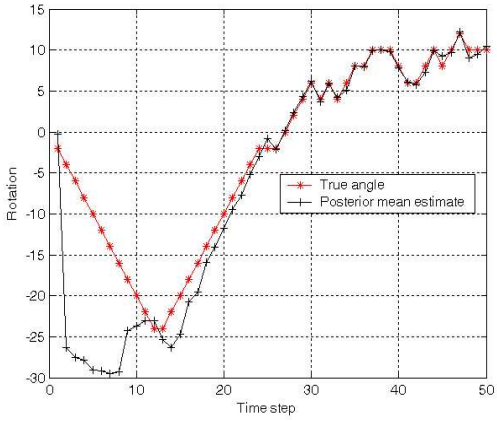
APF with initialization



APF without initialization

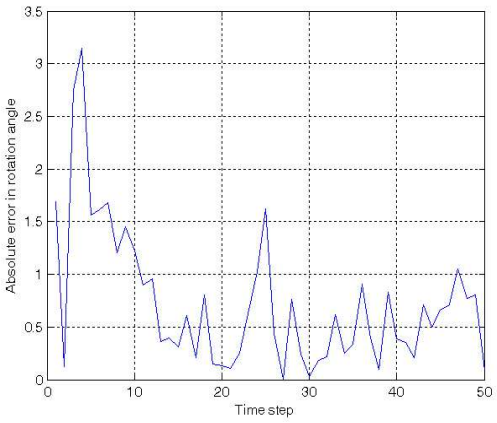


SIR with initialization

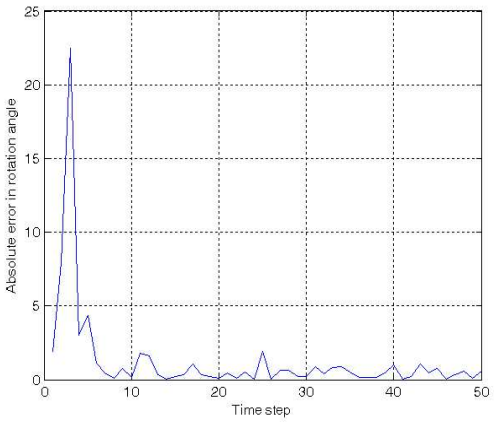


SIR without initialization

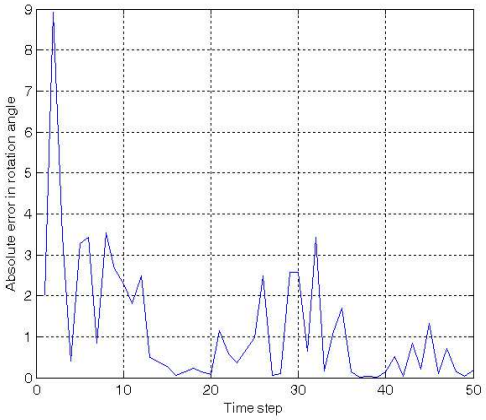
Figure 4.11: Rotation angle estimate



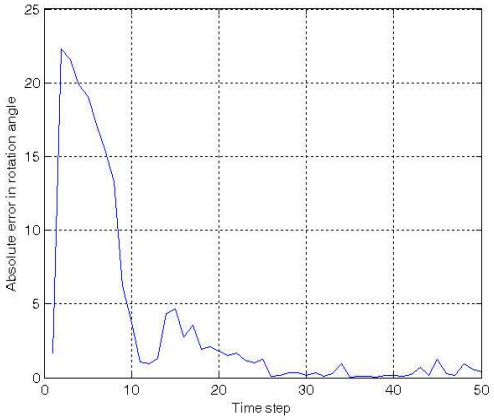
APF with initialization



APF without initialization

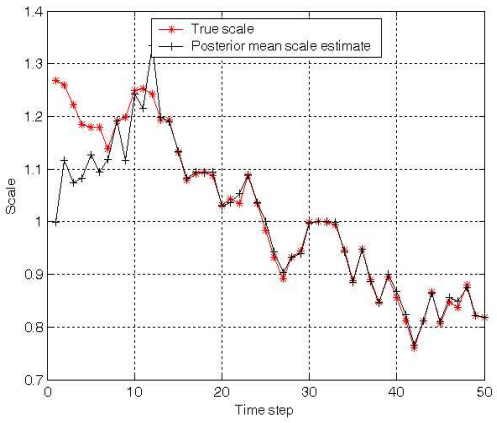


SIR with initialization

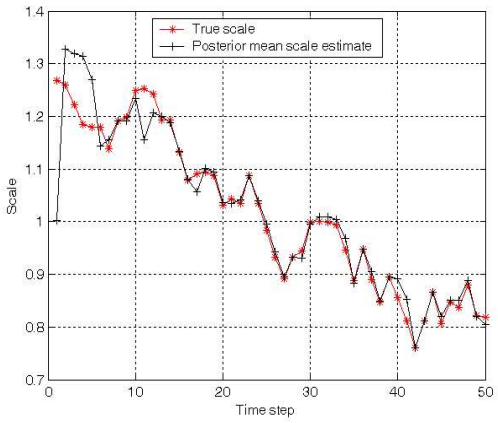


SIR without initialization

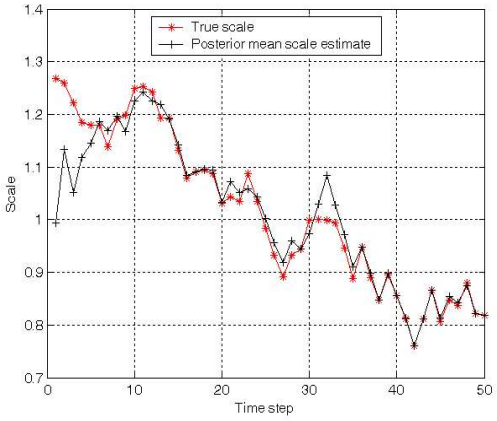
Figure 4.12: This figure and Fig. 4.11 show the rotation angle estimate and the corresponding absolute error respectively. As seen from the figures we observe that all the methods have a significant peak error in the first few steps of the angle estimation. The APF filter with initialization performs the best with a peak error of 3 degrees and the subsequent error in the angle is below 1 degree most of the time. The SIR filter with initialization has a peak error of about 9 degrees and then decreases significantly to about 1 degree. The angle estimation with this filter is not as smooth as of that in the APF with initialization case. Both the methods without initialization have the same peak error of about 22 degrees which is quite high, but they converge and begin tracking the correct angle, with the APF method converging before the SIR method. The absolute value of the error in the two methods after converging is about 2 degrees.



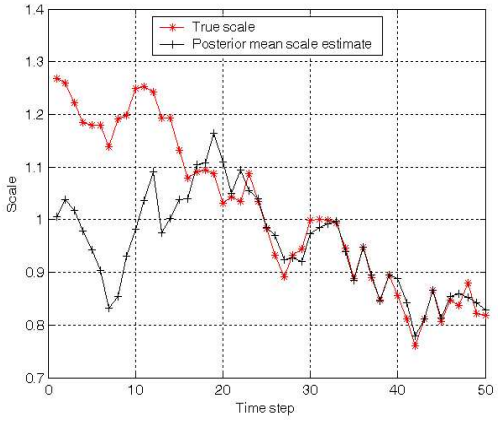
APF with initialization



APF without initialization

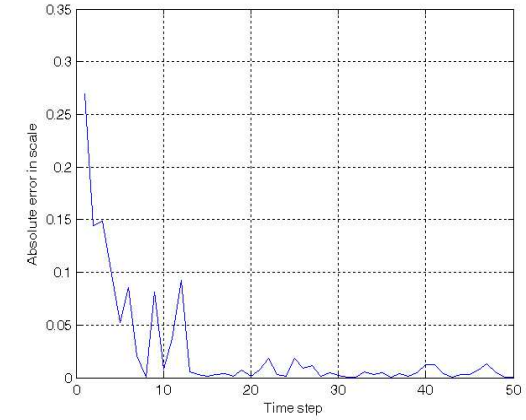


SIR with initialization

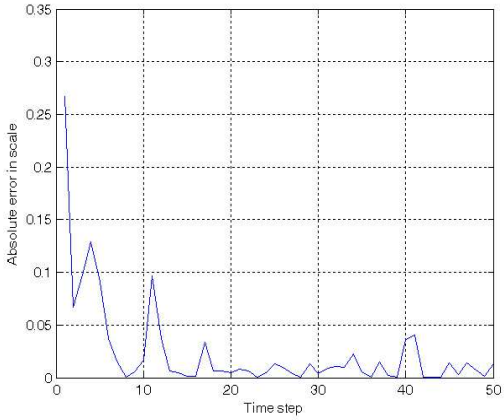


SIR without initialization

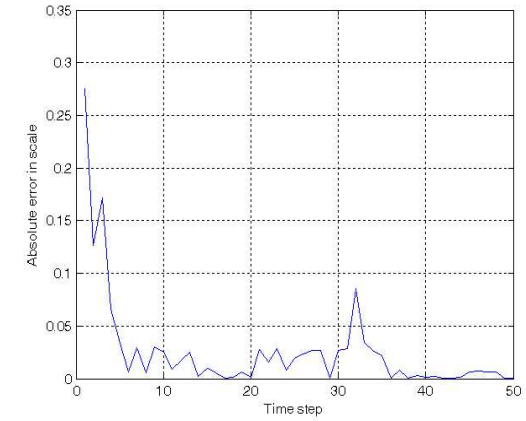
Figure 4.13: Scaling factor estimate



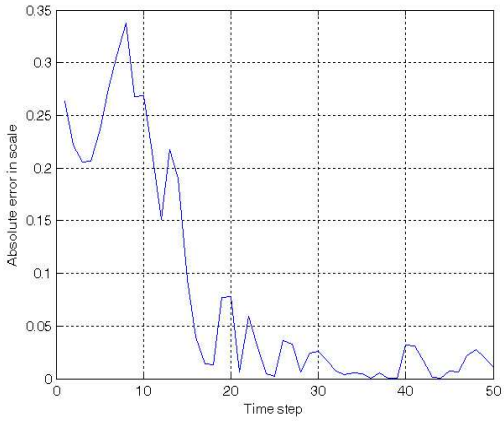
APF with initialization



APF without initialization



SIR with initialization



SIR without initialization

Figure 4.14: This figure and Fig. 4.13 show the scaling factor estimate and the corresponding error respectively. In this comparison the APF and SIR with initialization both converge comparably but the APF keeps the error below 0.05 consistently unlike the SIR filter. As seen in the previous cases here also the methods without initialization take considerable longer time to converge, with the SIR method taking the longest time to converge. After convergence the APF and SIR methods without initialization perform almost similarly

CHAPTER 5

Conclusions and Future Work

5.1 Conclusion

A nonlinear Bayesian algorithm based on sequential importance sampling was developed, which was able to track a moving target with random scale and rotation angles. The new algorithm added affine parameters to model the scale and rotational aspect of the target at every time step, and these parameters were estimated at every time step using particle filters. So now even though only a base template of the target has to be stored in the template library , we can track many different aspects of the target. Simulations show very good tracking performance for the APF tracker, which takes into consideration the current available observation, using 3500 particles. The SIR tracker also produces fairly good results. The first step of initializing particles according to a possibility map helps both the SIR and APF algorithms to converge faster with better tracking results. Even in cases with low SNR the estimated trajectory is quite accurate with the APF, but it does take a longer time to converge than the cases with higher SNR.

5.2 Future work

We propose to extend the framework to account for real multi aspect tracking with only a few base templates. By real multi aspect tracking we hope to track a real moving target given only a few of its possible signatures. The current algorithm can track only objects signature varying from one view but we believe the framework can be further extended to include all possible views of a 3D object. One idea to do this is by principle component

analysis (PCA), and using the state variables in the particle filter as eigen values of the PCA analysis. Using this approach we may be even able to track multiple objects moving in the same scene.

Secondly the current scaling and rotation models need to be made more realistic and also include the shearing factor more effectively. Also the likelihood function may be modified so that the information conveyed by the velocity component is also used in the likelihood computation.

Thirdly more recent advances in particle filtering technique, which include the likes of turbo particle filter [16], density assisted particle filters [17] may be incorporated into the algorithm to provide for more improvement in the tracking results.

BIBLIOGRAPHY

[1] Y. Bar-Shalom and X. Li., *Multitarget-Multisensor Tracking: Principles and Techniques*. CT: SYBS. Storrs, 1995.

[2] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 28, pp. 5–28, 1998.

[3] M. G. S. Bruno, "Bayesian methods for multiaspect target tracking in image sequences," *IEEE Trans. Signal Processing*, vol. 52, pp. 1848–1861, 2004.

[4] A. Doucet, J. F. G. Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. Eds. New York: Springer-Verlag, 1st ed., 2001.

[5] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation,," vol. 140, pp. 107–113, 1993.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, pp. 174–188, 2002.

[7] J. MacCormick and A. Blake, "A probabilitic exclusion principle for tracking multiple objects," *Intl conf comput vision*, pp. 572–578, 1999.

[8] J. Carpenter, P. Clifford, and P. Fearnead, "Improved particle filter for nonlinear problems," *Proc Inst Elect EngRadar Sonar Navig*, 1999.

[9] N. Bergman, "Recursive bayesian estimation: Navigation and racking applications," *Ph.D. dissertation, Linkoping Univ, Linkoping, Sweden*, 1999.

[10] A. Doucet, "On sequential monte carlo methods for bayesian filtering," *Dept. Eng., Univ. Cambridge, UK, Tech. Rep.*, 1998.

[11] R. van der Merwe, A. Doucet, J. F. G. de Freitas, and E. Wan, "The unscented particle filter," *Adv. Neural. Inform. Process. Syst.*

[12] M. Pitt and N.Shephard, "Filtering via simulation: auxiliary particle filters," *J. Amer. Statistic. Assoc.*, vol. 94, pp. 590–599, 1999.

[13] M. G. S. Bruno and J. M. F. Moura, "Multiframe detection/tracking in clutter: optimal performance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, pp. 925–946, 2001.

[14] J. M. F. Moura and N. Balram, "Noncausal gauss markov random fields:parameter structure and estimation," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1333–1355, 1992.

[15] M. G. S. Bruno, "Sequential importance sampling filtering for target tracking in image sequences," *IEEE Signal Processing Letters*, vol. 10, pp. 246–249, 2003.

[16] Z. Chen, T. Kirubarajan, and M. R.Moreland, "Improved particle filtering schems for target tracking," *IEEE ICASSP*, vol. 4, pp. 145–148, 2005.

[17] P. M. Djurić, M. F. Bugallo, and J. Míguez, "Density assisted particle filters for state and parameter estimation," *IEEE ICASSP*, vol. 2, pp. 701–704, 2004.

VITA

Vijay Venkataraman

Candidate for the Degree of

Master of Science

Thesis: ROTATION AND SCALING INVARIANT TARGET TRACKING USING PARTICLE FILTERS IN INFRARED IMAGE SEQUENCES

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in India, on November 21, 1981.

Education: Received the B.Tech degree from Indian Institute of Technology Madras Chennai, India, in 2003, in Electrical Engineering; Completed the requirements for the Master of Science degree with a major in Electrical Engineering at Oklahoma State University in July 2005.

Experience: Research Assistant at Oklahoma State University from January 2004 to July 2005.

Professional Memberships: IEEE Student Member.

Name: Vijay Venkataraman Date of Degree: July, 2005

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: ROTATION AND SCALING INVARIANT TARGET TRACKING USING PARTICLE FILTERS IN INFRARED IMAGE SEQUENCES

Pages in Study: 54 Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

A new rotation and scaling invariant target tracking algorithm is proposed using particle filters. Specifically, the target aspect is modeled by a continuous-valued affine model which is augmented to the target's kinematic parameters and whose dynamics are assumed to follow a first-order Markov model. Two specific particle filtering algorithms are implemented, i.e., Sequential Importance Re-sampling (SIR) and Auxiliary Particle Filter (APF). The Gaussian-Markov Random Field (GMRF) is used to characterize the spatial clutter of the background, and a target signature model is used to simulate the presence of a target. Simulation results show good tracking performance on targets with time varying rotation angles and scale factors even under low signal-to-noise ratios.

ADVISOR'S APPROVAL: Dr Guoliang Fan