2012

Particle filtered modified compressed sensing and applications in visual tracking

Rituparna Sarkar
Iowa State University

Particle filtered modified compressed sensing and applications in visual tracking

by

Rituparna Sarkar

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:

Namrata Vaswani, Major Professor

Elia Nicola

Yan Bin Jia

Iowa State University

Ames, Iowa

2012

DEDICATION

I would like to dedicate this thesis to my parents without whose support I would not have been able to complete this work.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Namrata Vaswani for her guidance, and support throughout this research and the writing of this thesis. I would also like to thank my committee members for their suggestions and comments on my research and contributions to this work. I would also like to thank Samarjit Das for his guidance and help with my research. I would also like to thank my friends and family for their loving guidance and support during my stay in Ames.

ABSTRACT

The main idea of the thesis is to design an efficient tracking algorithm that is able to track moving objects in presence of spatial illumination variation. The state vectors constitute of the motion parameters and the illumination vectors. The illumination vector is designed as a sparse vector using the fact that the scene parameters (e.g. illumination) at any given instant, can have a sparse representation with respect to the basis i.e. only a few basis elements will contribute to the scene dynamics at each instant. The observation is the entire image frame. The non-linearity and the multimodality of the state-space necessitates the use of Particle Filter. The illumination vector along with motion makes the state-space large dimensional thus making the implementation of regular particle filter expensive. PF-MT has been designed to tackle this problem but it does not utilize the sparsity constraint and hence fails to detect the sparse illumination vector. So we design an algorithm that would use particle filter and importance sample on the motion or the 'effective space' and the mode tracking step of PF-MT is replaced by the Modified Compressed Sensing for estimating the 'residual space'. Simulation and also experiments with real video demonstrate the advantage of the proposed algorithm over other existing PF based algorithms.

CHAPTER 1. INTRODUCTION

Tracking is a useful statistical signal processing technique used to estimate a hidden state sequence from a sequence of noisy observation that satisfy the Hidden Markov Model assumption. A tracking algorithm recursively computes the posterior distribution at time t using the posterior at time $t-1$. In case of visual tracking it involves determining the position of a known object from a sequence of image frames using the information of the object position in the previous frame. Here, we study the problem of recursive, and causal, estimation of a time sequence of sparse spatial signals, with slowly changing sparsity patterns, as well as other unknown states, from a sequence of nonlinear and noise corrupted observations. In many practical applications, particularly those in video processing and computer vision, the unknown state can be split into a small dimensional part and a spatial signal (large dimensional part). The spatial signal is often well modeled as a sparse signal. For a long sequence, its sparsity pattern (the support set of the sparsity basis coefficients' vector) can change over time, although the changes are slow. Moreover, due to temporal dependencies, the nonzero signal values also change slowly over time. For tracking problems that require causally estimating a time sequence of hidden states, X_t, from nonlinear and non-Gaussian measurements, Y_t that satisfy the hidden Markov model assumption, the most common and efficient solution is to use a particle filter (PF). The PF provides a sequential Monte Carlo approximation to the posterior. It uses sequential importance sampling [10] along with a resampling step [12] to empirically estimate the posterior distribution, $\pi_{t|t}(X_t) := f_{X_t|Y_{1:t}}(x_t|y_{1:t})$, of the state X_t conditioned on all observations up to the current time, $Y_{1:t}$. Here $f_{X_t|Y_t}$ refers to the PDF of X given Y.

1.1 Motivation for new algorithm

In this work we introduce a solution called Particle filtered Modified-CS (PaFiMoCS) that is inspired by PF-MT (Particle Filter with Mode Tracker). The key idea of PaFiMoCS is to importance sample on the small dimensional state vector, while replacing importance sampling by slow sparsity change constrained posterior mode tracking for recovering the sparse spatial signal. For every importance sampled particle of the small dimensional state vector, one solves the regularized Modified-CS problem to recover the spatial signal and its support. The weighting step is designed appropriately according to the importance sampling principle [10].

We show how to design PaFiMoCS for tracking moving objects across spatially varying illumination changes. Extensive experiments on both simulated data as well as on real videos involving significant illumination changes demonstrate the superiority of the proposed algorithm as compared with existing PF based tracking algorithms.

1.1.1 Need for new algorithm

Since the state space dimension in our problems is usually very large, the original PF [12] will require too many particles for accurate tracking and hence becomes impractical to use. As explained in [6], the same is essentially true for most existing PF algorithms. Some of the efficient PFs such as PF-Doucet[10], Gaussian PF [15], Gaussian sum filters or Gaussian sum PF [16] also cannot be used for the following reason. The first two implicitly assume that the posterior conditioned on the previous state, is unimodal or is at least unimodal most of the time. The second two assume a linear, or at least, a unimodal, observation model. In our problem, the observation model is nonlinear and is such that it often results in a multimodal observation likelihood, e.g., as explained in [6], this happens due to background clutter for the illumination tracking problem. If, in addition, the state transition prior of the small dimensional state, e.g., the motion states, is broad, which is often the case, it will result in the posterior being multimodal. Moreover, if the nonlinearity is such that the state to observation mapping is not differentiable, then one cannot even find the mode of $f_{X_t|Y_{1:t},X_{t-1}}(x_t|y_{1:t},x_{t-1})$ and hence cannot even implement PF-Doucet. This is again true for the illumination problem.

Frequently multimodal observation likelihoods and the above non-differentiability also mean that the extended Kalman filter [29], the unscented Kalman filter [29], the interacting multiple mode filter or Gaussian mixture filters cannot be used [11]. Rao-Blackwellized PF (RB-PF) [22, 4] and PF with posterior mode tracking (PF-MT) algorithm [25] are two possible solutions for large dimensional tracking problems, however, neither can exploit the sparsity or slow sparsity pattern change of the spatial signal. In addition, RB-PF also requires that conditioned on the small dimensional state vector, the state space model be linear and Gaussian.

1.1.2 Application of PaFiMoCS

Here, we use a template-based tracking framework with a simple three-dimensional motion model, that only models x-y translation and scale, because it is simple to use and to explain our key ideas. This necessitates illumination tracking along with object motion.When the illumination is constant, the motion of a rigid object moving in front of a camera can be tracked using a three dimensional vector consisting of x-y translation and uniform scale or more generally using a six dimensional affine model as in Condensation [13]. In Condensation the use of a particle filter (PF) for tracking through multimodal observation likelihoods resulting from background clutter or occlusions has been demonstated. Now if illumination also changes over time and if different parts of the object experience different lighting conditions, then more dimensions get added to the state space. Even a simple model of illumination such as that used in [14, 6], which parameterizes illumination using a Legendre basis, requires a 3-7 dimensional basis to represent illumination accurately. But even a 7-dimensional basis will increase the total state space dimension to between 10 and 13. A key example of the above problem occurs in tracking moving objects across spatially varying illumination changes, e.g. persons walking under a tree (different lighting falling on different parts of the face at different times due to the leaves blocking or not blocking the sunlight); or indoor sequences with variable lighting in various parts of the room, either due to the placement of light sources, or due to sunlight coming in through the windows that illuminates certain parts of the room better than others. In all of these cases, one needs to explicitly track the motion (small dimensional part) as well as the illumination. The illumination model is often represented using the top few coefficients of the Legendre basis (basis of Legendre polynomials) [14, 28, 6]. For videos with significant spatiotemporal illumination variations, the projection of the illumination into the Legendre basis is modeled as being a sparse vector [21], with slow sparsity pattern change.

1.2 Thesis outline

In the thesis we first discuss the basics of Particle Filter and Compressed Sensing. In 3rd chapter we design the dynamical state-space models which can be divided into a smaller dimensional vector and a large-dimensional sparse vector and discuss the development of two efficient algorithms for recursively recostructing a sparse signal using regularized modified sensing facilitated by particle filter to predict the support and the smaller dimensional state space by importance sampling. In 4th chapter we show an application of the algorithm for tracking a joint motion-illumination model, where we desing the illumination as a sparse vector. Lastly we show show our experimenation results in simulation as well as in real video, which includes tracking of a person under changing illumination conditions. In the last chapter discuss the concllusions and other future directions

CHAPTER 2. Background

2.1 Particle Filtering

In this chapter we go through the basics of Particle filter and sequential Monte Carlo techniques for Bayesian filtering [10],[1]. In the later half we discuss the basics of Compressed Sensing and Modfiied Sensing [18], [27]. Then we briefly discuss about the Particle filter with Mode tracker [6], [7] and the drawbacks that necessitated the design of Particle Filtering with Modified Compressed Sensing algorithm. As a general definition of Particle filter, we can say, it is an estimation technique for detection of states which are latent or hidden, from a given noisy observation data.

2.1.1 Introduction to Bayesian Filtering

Consider the following state space model [10],

$$\begin{aligned} X_t &= h(X_{t-1}) + w_t \\ Y_t &= g(X_t) + v_t \end{aligned} \tag{2.1}$$

Here X_t denotes the states and Y_t denotes the observation of the current state with discrete time $t = 0,1.......,n$. In real-life applications, for example, the state can be the position of a target while the observation is the noisy sensor data about the current position and our goal could be to extract the true state information using the observations and the state dynamical model. The function $h(.)$ and $g(.)$ can be either linear or non-linear. The state sequences X_t ; $t = 1,2....,n$ are assumed to be hidden Markov process and Y_t ; $t = 1,2....,n$ are conditionally independent observations. The following are assumed to be known:

i. $p(X_0)$; the initial state distribution

ii. $p(X_t \mid X_{t-1})$; the state transition density

iii. $p(Y_t \mid X_t)$; the observation likelihood.

Here $p(.)$ denotes the probability density function. (i) and (ii) can be obtained from the distribution of the noise w_t and v_t. We consider w_t and v_t as idndependent and identically distributed (iid) which can either be Gaussian or non-Gaussian. For simplicity we consider these to be Gaussian. We denote $X_{1:t} \equiv (X_1, ..., X_t)$ and $Y_{1:t} \equiv (Y_1, ..., Y_t)$ as the state sequence and the observations upto time T respectively.

Our aim is to estimate:

(a) The joint posterior state distribution at time t i.e, $p(X_{1:t} \mid Y_{1:t})$ or its marginal $p(X_t \mid Y_{1:t})$

(b) Expectation of the form: $I_t = E_{p(X_t|Y_{1:t})}(f_X(X_t)) = \int f_X(X_t)p(X_t \mid Y_{1:t})d(X_t)$

Here X is a random variable distributed over the interval $[a, b]$ where $f_X(X)$ is a continuous PDF. The state space are assumed to have Hidden markov Model (HMM) i.e, X_t is a Markov process and Y_t for $t = 1......n$, are conditionally independent of the previous states and previous observations. i.e., $p(Y_t|X_{t-1}, Y_{1:t-1}) = p(Y_t|X_t)$ and under HMM assumptions $p(X_t|X_{1:t-1}) = p(X_t|X_{t-1})$. When the posterior can be assumed to be Gaussian and $h(.)$, $g(.)$ to be linear the same problem can be solved using Kalman filter [29]. Extended Kalman filter [29] can be used, if $g(.)$ is non-linear, but it still assumes the gaussianity of the posterior distribution. But in many practical problems the posterior can be non-Gaussian with non-linear $h(.)$ and $g(.)$. Under such circumstances, sequential Monte Carlo technique based particle filtering algorithm gives us a way to solve this posterior estimation problem.

2.1.2 Derivation of Particle Filter

In order to compute the expectation w.r.t the joint posterior distribution it is required to know, if it is possible to sample from $p(X_{1:t} \mid Y_{1:t})$. and if we have a closed form of the expression. In most real life situations it is not possible to sample from the posterior i.e, $p(X_{1:t} \mid Y_{1:t})$ nor does it have a closed form of expression. Bayesian importance sampling is a proceedure to tackle this problem. The key idea is to represent the required posterior pdf by a set of random samples with associated weights and to compute estimates based on these samples and the weights. The derivation is based on [10].

2.1.2.1 Bayesian Importance Sampling

Since it is impossible to sample from $p(X_{1:t} \mid Y_{1:t})$, we adopt an importance sampling approach. Since we do not have a closed form of the expression $p(X_{1:t} \mid Y_{1:t})$, it can be expressed in the following manner,

$$\begin{aligned} p(X_{1:t} \mid Y_{1:t}) &= \frac{p(X_{1:t}, Y_{1:t})}{p(Y_{1:t})} \\ &\propto p(X_{1:t}, Y_{1:t}) \end{aligned} \tag{2.2}$$

A recursion can be obtained as $p(X_{1:t} \mid Y_{1:t}) = p(X_{1:t-1} \mid Y_{1:t-1})p(Y_t \mid X_t)p(X_t \mid X_{t-1})$ with $p(Y_t \mid X_t)$ and $p(X_t \mid X_{t-1})$ known. Let us consider the importance density function to be $\pi(X_{1:t} \mid Y_{1:t})$ from which we are going to draw samples. We choose $\pi(.)$ in such a way that we can recursively compute its expression and it has a convenient closed form expression from which we can easily draw samples. Now we write the posterior expectation as:

$$\begin{aligned} I_t &= \int f_X(X_{1:t})p(X_{1:t} \mid Y_{1:t})d(X_{1:t}) \\ &= \frac{\int f_X(X_{1:t})p(X_{1:t}, Y_{1:t})d(X_{1:t})}{p(Y_{1:t})} \\ &= \frac{\int f_X(X_{1:t})p(X_{1:t}, Y_{1:t})d(X_{1:t})}{\int_{X_{1:t}} p(X_{1:t}, Y_{1:t})d(X_{1:t})} \\ &= \frac{\int f_X(X_{1:t})\frac{p(X_{1:t},Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}\pi(X_{1:t} \mid Y_{1:t})d(X_{1:t})}{\int_{X_{1:t}} \frac{p(X_{1:t},Y_{1:t})}{\pi(X_{1:t}|y_{1:t})}\pi(X_{1:t} \mid Y_{1:t})d(X_{1:t})} \\ &= \frac{E_\pi(.)[f(X_{1:t})\frac{p(X_{1:t},Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}]}{E_\pi(.)[\frac{p(X_{1:t},Y_{1:t})}{\pi(X_{1:t}|Y_{1:t})}]} \end{aligned} \tag{2.3}$$

We can now draw sample from $\pi(.)$ as $X^i_{1:t} \sim \pi(X_{1:t} \mid Y_{1:t})$, where $i = 1,....,N_{pf}$, the discretized version of $\pi(.)$, given as $\hat{\pi}(.)$ can be obtained and I_t can be determined as:

$$\begin{aligned} I_t &= \frac{\frac{1}{N}\Sigma^N_{i=1}f_X(X^i_{1:t})\tilde{w}^i_t}{\frac{1}{N}\Sigma^N_{i=1}\tilde{w}^i_t} \\ &= \Sigma^N_{i=1}f_X(X^i_{1:t})w^i_t \end{aligned} \tag{2.4}$$

where $\tilde{w}^i_t = \frac{p(X^i_{1:t},Y_{1:t})}{\pi(X^i_{1:t}|Y_{1:t})}$ and $w^i_t = \frac{\tilde{w}^i_t}{\Sigma^N_{j=1}\tilde{w}^j_t}$

The corresponding approximation to the joint posterior distribution is given as,

$$\hat{p}(X_{1:t}, Y_{1:t}) \approx \Sigma_{i=1}^{N} w_t^i \delta(X_{1:t} - X_{1:t}^i) \tag{2.5}$$

Here the w_t^i is defined as the normalised weight such that $\Sigma_i w_t^i = 1$.

We choose an importance function of the form:

$$\pi(X_{1:t} \mid Y_{1:t}) = \pi(X_{1:t-1} \mid Y_{1:t-1})\pi(X_{1:t} \mid X_{1:t-1}, Y_{1:t}) \tag{2.6}$$

Since $p(X_{1:t}, Y_{1:t}) = p(X_{1:t-1}, Y_{1:t-1})p(Y_t \mid X_t)p(X_t \mid X_{t-1})$, we can develop a recursive way of computing the importance weight as,

$$\begin{aligned} \tilde{w}_t^i &= \frac{p(X_{1:t}^i, Y_{1:t})}{\pi(X_{1:t}^i \mid Y_{1:t})} \\ &= \tilde{w}_{t-1}^i \frac{p(Y_t \mid X_t^i)p(X_t^i \mid X_{t-1}^i)}{\pi(X_t^i \mid X_{1:t-1}^i, Y_{1:t})} \end{aligned} \tag{2.7}$$

where $X_t^i \sim \pi(X_t^i \mid X_{1:t-1}^i, Y_{1:t})$ and $X_{1:t}^i = [X_{1:t-1}^i, X_t^i]$ Thus the estimates of the posterior distribution can be computed recursively starting with the initial distribution.

2.1.2.2 Choice of Importance Function

The choice of the importance function is very crucial as it minimises the variance of the importance weight conditional upon the selected trajectory and observations. The following methods described are based on [10].

1. *Optimal Importance Function*: The importance function can be chosen in various ways, the simplest form is to use the state transtion density as the importance function,i.e,

$$\begin{aligned} \pi(X_t^i \mid X_{1:t-1}^i, Y_{1:t}) &= p(X_t \mid X_{t-1}) \\ \text{This gives } \tilde{w}_t^i &= \tilde{w}_{t-1}^i p(y_t \mid X_t^i) \end{aligned} \tag{2.8}$$

   It can be shown that the optimal importance density is one which minimizes the variance of the importance weight conditioned upon the observations and previous state samples (under HMM conditions) and $\pi_{opt}(.) = p(X_t \mid X_{t-1}, Y_t)$.

2. *Importance Distribution Obtained by Local Linearizarion*: Here a scheme is presented by which a Gaussian importance function is derived whose parameters are evaluated using local linearisation i.e, which are dependent on the simulated trajectory. Let us consider the following model:

$$\begin{aligned} X_t &= h(X_{t-1}) + w_t,\ w_t \sim \mathcal{N}(0, \Sigma_w) \\ Y_t &= g(X_t) + v_t,\ v_t \sim \mathcal{N}(0, \Sigma_v) \end{aligned}$$

where $h(.)$ and $g(.)$ are differentiable. Performing an approximation up to first order of the observation equation, we get:

$$Y_t \simeq g(h(X_{t-1})) + \frac{\delta g(X_t)}{\delta X_t}|_{X_t=h(X_{t-1})}[(X_t - h(X_{t-1})) + w_t] \tag{2.9}$$

Though this equation is not markovian as 2.9 is dependent on X_{t-1}, the Gaussian importance function can be obtained as: $\pi(X_t \mid X_{t-1}, Y_t) \sim \mathcal{N}(\mathrm{m}_t,\Sigma_t)$; with mean $(m_t$ and covariance Σ_t.evaluated for each trajectory $i = 1,....,n_{pf}$ using the following formula:

$$\Sigma_t^{-1} = \Sigma_w^{-1} + [\frac{\delta g(X_t)}{\delta X_t}|_{X_t=f(X_{t-1})}]' \Sigma_v^{-1} \frac{\delta g(X_t)}{\delta X_t}|_{X_t=f(X_{t-1})} \tag{2.10}$$

$$\begin{aligned} m_t &= \Sigma_t(\Sigma_w^{-1} h(X_{t-1}) + [[\frac{\delta g(x_t)}{\delta X_t}|_{X_t=h(X_{t-1})}]' \Sigma_v^{-1}] \\ &\quad \times (Y_t - g(h(X_{t-1})) + \frac{\delta g(X_t)}{\delta X_t}|_{X_t=h(X_{t-1})} f(X_{t-1}))) \end{aligned} \tag{2.11}$$

The associated importance weight is calculated using the following equation:

$$\tilde{w}_t^i = \tilde{w}_{t-1}^i \frac{p(Y_t \mid X_t^i) p(X_t^i \mid X_{t-1}^i)}{\pi(X_t^i \mid X_{1:t-1}^i, Y_{1:t})} \tag{2.12}$$

3. *Importance Distribution by Local Linearizarion of the optimal importance function*: Here we assume a function $l(X_t)$=log $p(X_t \mid X_{t-1}, Y_t)$ such that $l(x_t)$ is twice diffrenciable. We define:

$$l'(X_t) = \frac{\delta l(X_t)}{\delta X_t}|_{X_t=X} \tag{2.13}$$

$$l''(X_t) = \frac{\delta^2 l(X_t)}{\delta X_t \delta X_t'}|_{X_t=X} \tag{2.14}$$

(2.15)

The point $\mathbf{x}$ is so chosen that,

$$\begin{aligned}
\mathbf{x} &= \arg\max_{X_t} \, l(X_t) \\
&= \arg\max_{X_t} \, log\,[p(X_t \mid X_{t-1}, Y_t)] \\
&= \arg\max_{X_t} \, log\,[p(X_t \mid X_{t-1})] \, + \, log[p(Y_t \mid X_t)] \\
&= \arg\min_{X_t} \, -log\,[p(X_t \mid X_{t-1})] - log\,[p(Y_t \mid X_t)] \qquad (2.16)
\end{aligned}$$

Also we assume that $l''(X_t)$ is negative definite, then $l(X_t)$ is concave. We set:

$$\begin{aligned}
\Sigma(\mathbf{x}) &= -l''(\mathbf{x})^{-\mathbf{1}} \qquad (2.17) \\
and\; m(\mathbf{x}) &= \Sigma(x) l'(\mathbf{x})
\end{aligned}$$

Then the importance function can be calculated as:

$$\pi(X_t \mid X_{t-1}, Y_t) = \mathcal{N}(X_t; m(\mathbf{x}) + \mathbf{x}, \mathbf{\Sigma_x})$$

Assuming the function $p(X_t \mid X_{t-1}, Y_t)$ is unimodal, $\mathbf{x}$ can be adopted as the mode of that function for which $l'(X_t) = 0$. Hence $m(\mathbf{x}) = \mathbf{0}$ and the importance function is:

$$\pi(X_t \mid X_{t-1}, Y_t) = \mathcal{N}(X_t; \mathbf{x}, \mathbf{\Sigma_x}) \qquad (2.18)$$

The associated importance weight is calculated using equation .

2.1.3 Resampling

To prevent the degenerency of weights in SIS (sequential importance sampling) algorithm [10], the next step of the particle filter involves resampling. The particles are resampled w.r.t their normalised importance weights i.e, $\{w_t^i\}_{i=1}^{N_{pf}}$ is used as the probability mass function to sample the existing particles again. The basic idea of resampling is to discard the particles that have small weights and concentrate on the particles that have larger weights. The resampling involves generating a new set $\{x_t^{i*}\}_{i=1}^{N_{pf}}$ such that $Pr\{x_t^{i*} = x_t^j\} = w_t^j$. The weights are then reset to $\frac{1}{N}$.

2.1.4 The Basic Particle Filter

The algorithm for the Particle Filter is given as [10]:

1. Initate: At time $t = 0$, sample $X_0^i \sim \mathcal{N}(0,\Sigma_0)$ for $i = 1.....n_{pf}$
2. For $t \geq 0$,
   a. Sample $\mathrm{X}_t^i \sim \pi(X_t \mid X_{t-1}^i, Y_t)$ for $i = 1.....N_{pf}$
   b. Assign the particle a weight, $\tilde{w}_t^i$, according to (2.7)
   c. Calculate the normalised weight $\mathrm{w}_t^i = \frac{\tilde{w}_t^i}{\Sigma_{j=1}^N \tilde{w}_t^j}$
   d. Resample particles as $\mathrm{X}_t^i \sim \mathrm{PMF}[\{\mathrm{w}_t\}]$ and Reassign $\mathrm{w}_t^i = \frac{1}{N}$
   e. Set the resampled particles $\mathrm{X}_{1:t}^i = [\mathrm{X}_{1:t-1}^i, \mathrm{X}_t^i]$
   f. Compute the posterior PDF as per (2.5) and the posterior expectation as per (2.4)
3. Set t+1 $\leftarrow$ t and go to step (2)

2.1.5 A Review of Particle Filter with Mode Tracker and its limitations

When the dimensionality of the state increases the two issues that are faced are that the observation likelihood becomes multimodal and the application of PF requires a large number of particles. Particle filter with efficient importance sampling (PF-EIS) was proposed in [25] to handle multimodal observation likelihoods with more details in [[26], [24], [23]]. Now, if the state-space dimensionality is large (10 or more), it makes particle filtering even more challenging because the number of particles required for reasonable accuracy in estimating the state becomes very large. Rao Blackwellization (RB-PF) [[22], [4]] can be used to handle this problem provided the state space model is conditionally linear-Gaussian. For many practical problem, this assumption does not hold. But in most large dimensional problems, the state change variance is large in only a few dimensions i.e. the LDSS property [8] holds i.e, at a given time the state change is large for only a few states and for the other states the change is small. Hence the design of PF-MT. The key idea of PF-MT is as follows [25]. It splits the state vector X_t into $X_t = [X_{t,s}, X_{t,r}]$ where $X_{t,s}$ denotes the coefficients of a small dimensional "effective basis" (in which most of the state change is assumed to occur) while $X_{t,r}$ belongs to the "residual space" in which the state change is assumed "small". It importance samples only on the effective basis dimensions, but replace importance sampling by deterministic posterior Mode Tracking (MT) in the residual space. Thus the importance sampling dimension is only $dim(X_{t,s})$ (much smaller than $dim(X_t)$) and this is what decides the effective particle size. PF-MT implicitly assumes (i) that the posterior of the residual space conditioned on the previous state and the effective basis ("conditional posterior") is unimodal most of the time; and that (ii) it is also narrow enough. Under these two assumptions, it can be argued that any sample from the conditional posterior is close to the conditional posterior mode with high probability [25, Theorem 2].

PF-MT can be directly applied to our problem if we do not use the sparsity of Λ_t. Then, with $X_{t,s} = U_t$ and $X_{t,r} = \Lambda_t$ we get the PF-MT algorithm given in Algorithm 1.

Algorithm 1 PF-MT: Particle Filter with posterior Mode Tracker

For all $t \geq 0$ do

1. For each particle i: Importance sample U_t from its prior: $U_t^i \sim \mathcal{N}(0, \Sigma_u)$
2. For each particle i: Mode track Λ_t: compute the mode of the posterior of Λ_t conditioned on X_{t-1}^i and U_t^i, i.e. compute Λ_t^i as the solution of
$$\min_{\Lambda} C(\Lambda) := -\log f_Z(Y_t - h([U_t^i, \Phi\Lambda])) + \frac{\|\Lambda - \Lambda_{t-1}^i\|_2^2}{2\sigma_l^2}$$
3. For each particle i: Compute the weights as follows.
$$w_t^i \propto w_{t-1}^i f_Z(Y_t - h([U_t^i, \Phi\Lambda_t^i]))\mathcal{N}(\Lambda_t^i; \Lambda_{t-1}^i, \sigma_l^2 I)$$
4. Resample and reset weights. Increment t and go to step 1.

However, since PF-MT does not exploit the sparsity or slow sparsity pattern change of Λ_t, it results in a dense solution for Λ_t, i.e. the energy gets distributed among all components of Λ_t. This becomes a problem in applications where Λ_t is indeed well approximated by a sparse vector with changing sparsity patterns. An alternative could be to assume selected fixed subset of Λ_t, i.e. fix $T_t = T_0$. For example, if Φ is a Fourier basis or a Legendre basis, one would pick the top few components as the set T_0. This was done in [6] for illumination. This approach works if most energy of L_t does indeed lie in the lower frequency (or lower Legendre) components, but fails if there are different types of high-frequency spatial variations in L_t over time. We demonstrate this for the illumination problem later.

2.2 Compressed Sensing

Suppose we have to reconstruct a sparse signal x from the measurement: $y = \Phi x$ when $n =$ length(y) , $m =$ length(x) and $n < m$ and Φ is an $m \times n$ matrix. We can say that y is given as the inner product of x and a collection of vectors $\sum_{(j=1)}^{m} \Phi_j$. The problem consists of designing

i. A stable measurement matrix Φ

ii. A reconstruction algorithm to recover x with support N.

i. Since $m < n$ the problem appears to be ill conditioned. But if x is N-sparse and the non-zero coefficients of x are known, then the problem can be solved provided $m \geq N$. A necessary and sufficient condition for this simplified problem to be well conditioned is that for any vector v sharing the same N coefficeints for some $\epsilon > 0$ [3].

$$(1 - \epsilon) \leq \frac{\|\Phi \nu\|_2}{\|\nu\|_2} \leq (1 + \epsilon) \tag{2.19}$$

Which means the matrix Φ must preserve the lengths of these particulat T-sparse vectors. Direct construction of Φ requires $\binom{n}{N}$ possible combinations of T non-zero entries of vector. So in practice if each element of Φ is chosen to be iid random variables to have the RIP (restricted isometry property given by equation (2.19)) with high probability.

ii. The classical approach to inverse problems of this type is to find the vector with the smallest l_2 norm by solving:

$$\begin{aligned} \hat{x} &= \arg\min_x \|x\|_2 \\ s.t,\ y &= \phi x \end{aligned} \tag{2.20}$$

This has a convenient closed form solution $\hat{x} = \Phi^\top (\Phi \Phi^\top)^{-1} y$. But this would never yield a sparse solution but instead would give a nonsparse $\hat{x}$ with many non-zero values. While

l_2 norm measures signal energy and not sparsity, l_0 norm on the other hand counts the number of non-zero entries. Hence the modified optimization problem is given by:

$$\begin{aligned} \hat{x} &= \arg\min_x \|x\|_0 \\ s.t,\ y &= \phi x \end{aligned} \tag{2.21}$$

But solving this is numerically unstable as well as NP-complete requiring an exhaustive search of all possible nonzero entries [19]. On the other hand the l_1 norm, can exactly recover sparse signals as suggested in [[9], [5], [2]], given as:

$$\begin{aligned} \hat{x} &= \arg\min_x \|x\|_1 \\ s.t,\ y &= \phi x \end{aligned} \tag{2.22}$$

2.2.0.1 Modified Compressed Sensing

Now if only a partial knowledge of the support is known, a modified version of compressive sensing can reconstruct the signal from even lesser number of measurements compared to traditional CS. The key idea of mod-cs is [[27]], given a partial but partly erroneous support knowledge: T, we can write the support of x as $N = T \cup \Delta \setminus \Delta_e$, where $\Delta = N \setminus T$ is the unknown set of misses in T and $\Delta_e = T \setminus N$ is the unknown set of extras in T. Now if $N = T \cup \Delta$, the CS problem reduces to finding a solution that is sparsest on T^c. Hence modified CS attempts to solve:

$$\begin{aligned} \hat{x} &= \arg\min_x \|x_{T^c}\|_1 \\ s.t,\ y &= \phi x \end{aligned} \tag{2.23}$$

Dynamic Modified sensing: One of the applications of modified CS is recursive reconstruction of sparse signals with time i.e. the CS algorithm becomes dynamic. For a time sequence we solve equation (2.23) with $T = \hat{N}_{(t-1)}$, where $\hat{N}_{(t-1)}$ is the estimate of the support from $t-1$ given as: $\hat{N} = \{i \in [1, n] : \hat{x}_i^2 \geq \alpha\}$. where $\alpha \geq 0$ is the zeroing threshold. The threshold is so chosen that α is slightly equal to or slightly less that the smallest value of the support, so that it ensures zero misses and very few false additions. For compressed signals, α would be evaluated after replacing the support with $b\%$-support, which means it would contain $b\%$ of the signal energy. For the noisy observation case it becomes the modified BPDN [27]

$$\hat{x}_t = \arg\min_x \|x_{T^c}\|_1 + \|y_t - \phi x\|_2^2 \tag{2.24}$$

2.2.0.2 Regularised Modified Compressed Sensing

Now if we have the knowledge of how the signal x_T was generated, we can use this to reduce the reconstructional error by solving [17]:

$$\begin{aligned} \hat{x} &= \arg\min_x [\gamma \|x_{T^c}\|_1 + \|x_T - \mu_T\|_2^2] \\ y &= \Phi x \end{aligned} \tag{2.25}$$

Where μ_T is the mean of the distribution of x_T. *Dynamic Regularized Modified compressed sensing*: For a time sequence of signals, we can apply equation (2.25) with $T = \hat{N}_{t-1}$ and $\mu_T = (\hat{x}_(t-1))_T$ and hence solve [17] :

$$\begin{aligned} \hat{x}_t &= \arg\min_x [\gamma \|x_{T^c}\|_1 + \|x_T - (x_{t-1})_T\|_2^2] \\ y &= \Phi x \end{aligned} \tag{2.26}$$

For the noisy case, when the noise is large this extra constraint put in by the Regularised Modified CS becomes more effective and is given by [17],

$$\hat{x}_t = \arg\min_x [\gamma \|x_{T^c}\|_1 + \|x_T - (x_{t-1})_T\|_2^2 + \|y_t - \phi x\|_2^2] \tag{2.27}$$

The Modified Compressed Sensing works when the support set of x_t changes slowly over time and also the change in the values of x are small.

CHAPTER 3. Particle Filtered Modified Compressed Sensing

The key idea of PaFiMoCS is to importance sample on the small dimensional state vector, while replacing importance sampling by slow sparsity change constrained posterior mode tracking for recovering the sparse spatial signal. For every importance sampled particle of the small dimensional state vector, one solves the regularized Modified-CS problem to recover the spatial signal and its support. The weighting step is designed appropriately according to the importance sampling principle [10]. We successfully demonstrated the use of PF-MT for visual tracking across certain types of illumination variations in [6]. But as, PF-MT also does not exploit the sparsity or slow sparsity pattern change of Λ_t. In situations where Λ_t is well approximated by a sparse vector whose support set does change over time, as explained in the introduction, most existing PF algorithms cannot be used for our problem, since we would like to (a) deal with multimodal observation likelihoods, (b) large dimensional state spaces and (c) the state being a sparse spatial signal with unknown and slowly changing sparsity patterns. However as we explain below the PF-MT idea can be adapted to solve this problem. It is possible to modify PF-MT to also utilize sparsity and slow sparsity pattern change and doing this removes the limitation of PF-MT explained above. The main idea is to use regularized Modified-CS proposed earlier for linear problems with slow sparsity pattern and signal value change in the mode tracking step of PF-MT. We refer to the resulting algorithm as Particle Filtered Modified-CS (PaFiMoCS) [21]. In situations where Λ_t is indeed well approximated by a sparse vector with a changing sparsity pattern, this significantly improves reconstruction performance. We demonstrate this for the illumination application later. We show how to design PaFiMoCS for tracking moving objects across spatially varying illumination changes. Extensive experiments on both simulated data as well as on real videos involving significant illumination changes demonstrate the superiority of the proposed algorithm as compared with existing PF based tracking algorithms.

3.1 Notation

The notation $||b||_k$ is used to denote the l_k norm of vector b. For any set T and vector b, $(b)_T$ is used to denote a subvector containing the elements of b with indices in T. For a matrix A, $(A)_T$ denotes the submatrix by extracting columns of A with indices in T. The term $(x_t)_{N_t}$ denotes the vector comprising of the elements of x_t with indices N_t at time t. We denote the complement set as N_t^c i.e., the set of indices of x_t that do not belong to N_t. The symbol $'\backslash'$ denotes the set difference. While going from $t-1$ to t, the set of new elements to be added are denoted by S_t whereas the set of deleted elements be denoted by R_t. Thus $S_t = N_t \setminus N_{t-1}$ and $R_t = N_{t-1} \setminus N_t$. The $\cup$ and $\cap$ denote set-union and set-intersection respectively. For a set N_t, $|N_t|$ denote the cardinality of a set, but for a scalar x, $|x|$ denotes the magnitude of x. The notation $vec(.)$ denote vectorization operation which operates on a matrix $m \times n$ to give a vector of size mn by cascading the rows. The Hadamard product is denoted by $\odot$. The function $round(.)$ operates on a matrix $\mathbf{Z}$ to output a matrix with integer entries closest to $z_{i,j} \forall i,j$ and the operator $mean(.)$ gives the arithmetic mean of a vector. The notation $\mathcal{N}(y;\mu,\Sigma)$ denotes the value of Gaussian distribution with mean μ and covariance Σ computed at y and $x \sim \mathcal{N}(\mu,\Sigma)$ denotes that random variable x is Gaussian distributed with mean μ and covariance Σ. Similarly, the notation $\mathcal{U}(a;c_1,c_2)$ denotes the value of the uniform density defined over $[c_1,c_2]$ computed at a while $x \sim \mathcal{U}(c_1,c_2)$ denotes that x is uniformly distributed over $[c_1,c_2]$. The notation $Unif_p(N)$ selects randomly any p unique entries of N where $p \ll |N|$. $S \sim Ber(N,p)$ means each particle in N has a probability p of being present in S and are independent of each other. $\mathbf{I}$ denotes the identity matrix. The terms $\mathbf{1}$ and $\mathbf{0}$ refer to column vectors with all entries as 1 and 0 respectively. $A\top$ denotes the transpose of a vector/matrix.

3.2 Problem Formulation

The goal is to recursively recover a time sequence of states X_t from noise-corrupted and nonlinear measurements, Y_t, when the state vector X_t can be split into two parts, a large dimensional, L_t, and a small dimensional part, U_t, with the following properties

1. L_t is in fact a spatial signal, that is sparse (many elements of L_t or of a linear transform of L_t are zero)

2. the sparsity pattern of L_t changes slowly over time and the same is true for its nonzero coefficients

Mathematically, this means the following. The observation Y_t satisfies

$$Y_t := h(X_t) + Z_t, \; Z_t \overset{\text{i.i.d.}}{\sim} f_Z(z) \tag{3.1}$$

i.e. Z_t is independent and identically distributed (i.i.d.) observation noise with probability density function (pdf) at any time given by $f_Z(z)$. In many situations, this is Gaussian. However, often to deal with outliers, one models Z_t as a mixture of two Gaussian pdf's, one which has small variance and large mixture weight and the second with large variance but small mixture weight. For the above model, the observation likelihood, $\mathrm{OL}(X_t)$, can be written as

$$\mathrm{OL}(X_t) := f_{Y_t|X_t}(Y_t|X_t) = f_Z(Y_t - h(X_t)) \tag{3.2}$$

More generally, sometimes the observation model is specified implicitly, i.e. it can only be written in the form

$$\tilde{h}(Y_t, X_t) = Z_t, \; Z_t \overset{\text{i.i.d.}}{\sim} f_Z(z) \tag{3.3}$$

In this case, the observation likelihood, $\mathrm{OL}(X_t)$, becomes

$$\mathrm{OL}(X_t) = f_Z(\tilde{h}(Y_t, X_t)) \tag{3.4}$$

Notice that (3.1) is a special case of (3.3) with $\tilde{h}(Y_t, X_t) = Y_t - h(X_t)$.

The state X_t can be split as

$$X_t = \begin{bmatrix} U_t \\ L_t \end{bmatrix}$$

where L_t is a sparse signal, i.e. it can be rewritten as

$$L_t = \Phi\Lambda_t \tag{3.5}$$

and Λ_t is a n_l-length sparse vector with support set T_t, i.e.

$$T_t := \text{support}(\Lambda_t) = \{j : (\Lambda_t)_j \neq 0\} \tag{3.6}$$

We assume the following models on T_t and $(\Lambda_t)_{T_t}$ and on U_t.

$$\begin{aligned} T_t &= T_{t-1} \cup A_t \setminus R_t, \text{ where} \\ A_t &\sim \text{Ber}(T_{t-1}^c, p_a) \\ R_t &\sim \text{Ber}(T_{t-1}, p_r) \end{aligned} \tag{3.7}$$

$$\begin{aligned} (\Lambda_t)_{T_t} &= (\Lambda_{t-1})_{T_t} + (\nu_{l,t})_{T_t}, \; (\nu_{l,t})_{T_t} \sim \mathcal{N}(0, \sigma_l^2 I) \\ (\Lambda_t)_{T_t^c} &= 0 \end{aligned} \tag{3.8}$$

Slow support change means that p_a and p_r are small. Slow signal value change means that σ_l^2 is small.

In the absence of any other specific information, we also assume a linear Gaussian random walk model on U_t.

$$U_t = U_{t-1} + \nu_{u,t}, \; \nu_{u,t} \sim \mathcal{N}(0, \Sigma_u) \tag{3.9}$$

If the only thing that is known is that the values of $(\Lambda_t)_{T_t}$ and U_t change slowly, then the above linear Gaussian random walk model is the most appropriate one. However, as far as the proposed algorithms are concerned, they are also applicable with minor changes for the case where $(\Lambda_t)_{T_t} = g(\Lambda_{t-1}, \nu_{l,t})$ and $g(.)$ is known.

The state transition prior, STP, corresponding to the above system model can be written out as follows.

$$\begin{aligned} \text{STP}(X_t^i; X_{t-1}^i) &:= f_{X_t|X_{t-1}}(X_t^i | X_{t-1}^i) \\ &= \text{STP}(T_t^i; T_{t-1}^i)\text{STP}(\Lambda_t^i; \Lambda_{t-1}^i, T_t^i) \times \\ &\quad \text{STP}(U_t^i | U_{t-1}^i) \end{aligned} \tag{3.10}$$

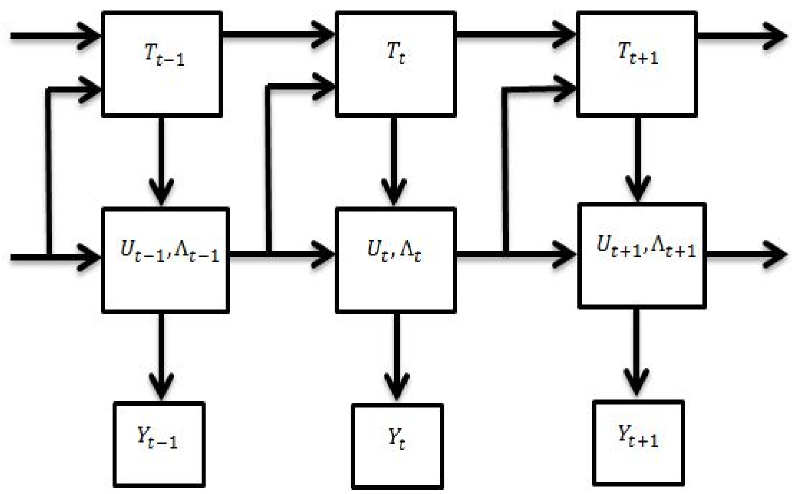

Figure 3.1 Markov Model of givel state space model

where

$$
\begin{aligned}
\mathrm{STP}(T_t^i; T_{t-1}^i) &:= P(T_t = T_t^i | T_{t-1} = T_{t-1}^i) \\
&= P(A_t = (T_t^i \setminus T_{t-1}^i), R_t = (T_{t-1}^i \setminus T_t^i)) \\
&= p_a^{|T_t^i \setminus T_{t-1}^i|}(1-p_a)^{n_l - |T_{t-1}^i| - |T_t^i \setminus T_{t-1}^i|} \times \\
&\quad\; p_r^{|T_{t-1}^i \setminus T_t^i|}(1-p_r)^{|T_{t-1}^i| - |T_{t-1}^i \setminus T_t^i|} \qquad (3.11) \\
\mathrm{STP}(\Lambda_t^i; \Lambda_{t-1}^i, T_t^i) &:= f_{\Lambda_t|\Lambda_{t-1}, T_t}(\Lambda_t^i | \Lambda_{t-1}^i, T_{t-1}^i) \\
&= \mathcal{N}((\Lambda_t^i)_{T_t^i}; (\Lambda_{t-1}^i)_{T_t^i}, \sigma_l^2 I) \qquad (3.12) \\
\mathrm{STP}(U_t^i; U_{t-1}^i) &:= f_{U_t|U_{t-1}}(U_t^i | U_{t-1}^i) \\
&= \mathcal{N}(U_t^i; U_{t-1}^i, \Sigma_u) \qquad (3.13)
\end{aligned}
$$

the markov model for our designed problem is shown by Fig. (3.1).

3.2.1 PaFiMoCS: *Particle Filtered Modified-CS algorithm*

For PaFiMoCS, we let $X_{t,s} = U_t$ and we let $X_{t,r} = [T_t, \Lambda_t]$. In the cost function that we minimize for the mode tracking step, we also include a term of the form $\|\Lambda_{T^c}\|_1$ with $T = T_{t-1}^i := \{j : |(\Lambda_{t-1}^i)_j| > \alpha\}$, i.e. T is an estimate of the support of Λ_{t-1}^i computed using a threshold $\alpha \geq 0$. Doing this is a tractable approximation to trying to find the vector Λ that is sparsest outside the set T (i.e. the vector with the smallest number of new support additions to T) among all vectors Λ that satisfy the observation model constraint (often referred to as the data constraint) and are "close enough" to the previous estimate, $(\Lambda_{t-1}^i)_T$. Also, since T_t is part of the residual state space, we need to include a term proportional to its state transition prior in the weighting step.

Algorithm 2 PaFiMoCS: Particle Filtered Modified-CS

Input: Y_t
Output: $U_t^i, T_t^i, \Lambda_t^i, w_t^i$
Parameters: (algorithm) α, γ, (model) $\Sigma_u, \sigma_l^2, p_a, p_r, f_Z(z)$
For all $t \geq 0$ do

1. For each particle i: Importance sample U_t from its prior: $U_t^i \sim \mathcal{N}(0, \Sigma_u)$
2. For each particle i: Mode track Λ_t, T_t with imposing slow sparsity pattern change, i.e. compute Λ_t^i as the solution of

$$\min_{\Lambda} C(\Lambda) := -\log f_Z(Y_t - h([U_t^i, \Phi\Lambda])) + \frac{\|\Lambda - \Lambda_{t-1}^i\|_2^2}{2\sigma_l^2} + \gamma\|\Lambda_{T^c}\|_1$$
$$\text{and } T = T_{t-1}^i$$

   and compute T_t^i by thresholding Λ_t^i, i.e.

$$T_t^i := \{j : |(\Lambda_t^i)_j| > \alpha\}$$

3. For each particle i: Compute the weights as follows

$$w_t^i \propto w_{t-1}^i f_Z(Y_t - h([U_t^i, \Phi\Lambda_t^i]))\mathcal{N}(\Lambda_t^i; \Lambda_{t-1}^i, \sigma_l^2 I)\mathrm{STP}(T_t^i; T_{t-1}^i)$$

   where $\mathrm{STP}(T_t^i; T_{t-1}^i)$ is defined in (3.11).
4. Resample and reset weights. Increment t and go to step 1.

3.2.2 PaFiMoCS-support: *PaFiMoCS for faster support changes*

A second approach which is useful when support changes of Λ_t are faster is to also include T_t as part of the state on which we importance sample, i.e. to use $X_{t,s} = [U_t, T_t]$ and $X_{t,r} = \Lambda_t$. The resulting algorithm is summarized in Algorithm 3.

Algorithm 3 PaFiMoCS-support: PaFiMoCS for faster support changes

Input: Y_t

Output: $U_t^i, T_t^i, \Lambda_t^i, w_t^i$

Parameters: (algorithm) α, γ, (model) $\Sigma_u, \sigma_l^2, p_a, p_r, f_Z(z)$

For all $t \geq 0$ do

1. For each particle i: Importance sample U_t from its prior: $U_t^i \sim \mathcal{N}(0, \Sigma_u)$

2. For each particle i: Importance sample T_t from its prior: $T_t^i = T_{t-1}^i \cup A_t^i \setminus R_t^i$ where $A_t^i \sim \text{Ber}((T_{t-1}^i)^c, p_a)$ and $R_t^i \sim \text{Ber}(T_{t-1}^i, p_r)$.

3. For each particle i: Mode track Λ_t, T_t with imposing slow sparsity pattern change, i.e. compute Λ_t^i as the solution of

$$\min_{\Lambda} C(\Lambda) := -\log f_Z(Y_t - h([U_t^i, \Phi\Lambda])) + \frac{\|\Lambda - \Lambda_{t-1}^i\|_2^2}{2\sigma_l^2} + \gamma \|\Lambda_{T^c}\|_1$$
$$\text{and } T = T_t^i$$

   Update T_t^i as

$$T_t^i := \{j : |(\Lambda_t^i)_j| > \alpha\}$$

4. For each particle i: Compute the weights as follows

$$w_t^i \propto w_{t-1}^i f_Z(Y_t - h([U_t^i, \Phi\Lambda_t^i]))\mathcal{N}(\Lambda_t^i; \Lambda_{t-1}^i, \sigma_l^2 I)$$

5. Resample and reset weights. Increment t and go to step 1.

CHAPTER 4. Visual Tracking Across Partially Varying Ilumination Changes

In this section, we focus on visual tracking across spatially varying illumination changes which is an important practical example of the general problem studied above. Here we express a spatial illumination variation as a sparse vector. For our problem we represent the illumination patterns by a sufficiently large dimensinal Legender basis functions so that more complex illumination patterns, which can be represented by the higher order Legender basis coefficient can also be accomodated [21]. However that would lead to to a large dimensional residual space and using this fact we can have a sparse representation of a complex illumination pattern, i.e, at a certain time only a few Legendre basis coefficients will contribute towards definig the illumination and the others would either be zero or have insignificantly small values. Considering the visual tracking problem we can split our state vector into the large dimensional Legendre basis coefficients Λ_t and the small dimensional motion parameters U_t. Now in case of spatially varying illuminatin change. We show in this section that the legende basis coeffients are sparse and also the sparsity pattern changes slowly over time.

4.1 State Space Model

The system model consists of simple dynamical models for illumination, Λ_t and motion parameters U_t and the suppoprt of the illumination model T_t. The observation Y_t is given as the image frame at time t.

4.1.0.1 System Model

We model the motion parameter U_t such that it consists of scale, horizontal translation and vertical translation and is given as $U_t = [s_t, u_t^x, u_t^y]^T$ such that its dimension is limited to 3 and $\Lambda_t \in R^D$ are the coefficients of the Legendre basis function. Both U_t and Λ_t follow the model as defined in the previous section given by (3.9), (3.8) and the support change model of Λ_t is given by (3.7). Also the corresponding state transition prior (STP) are as defined by (3.11), (3.12), (3.13).

4.1.1 Observation Model

We use the observation model similar to [6]. The changed appearance of the image at time t, I_t is represented in terms of a linear combination of the initial template I_0 scaled by a set of Legendre basis functions as introduced in [14]. Let p_k denote the k^{th} Legendre basis function, then with $D = 2k+1$, the template I_t is computed as follows:

$$vec(I_t) = \Phi\Lambda_t + I_0 \tag{4.1}$$

The matrix Φ has its columns consisting of the initial template scaled by D legendre basis functions and is defined as [6] :

$$\Phi = [vec((I_0) \odot p_0), ..., vec((I_0) \odot p_{D-1})] \tag{4.2}$$

Λ_t, a $D\times 1$ vector, is the Legendre basis coefficients at time t and hence is called the *illumination vector* and $\mathbf{P}$, the Legendre basis matrix is as defined in [[14, 6]] given as:

$$\mathbf{P} \triangleq \begin{bmatrix} p_0 & p_1(x_1) & \cdots & p_k(x_1) & \cdots & p_k(y_1) \\ \vdots & \vdots & \vdots & \vdots & & \\ p_0 & p_1(x_1) & \cdots & p_k(x_1) & \cdots & p_k(y_M) \\ \vdots & \vdots & \vdots & \vdots & & \\ p_0 & p_1(x_M) & \cdots & p_k(x_M) & \cdots & p_k(y_M) \end{bmatrix} \tag{4.3}$$

Here D relates to n_l defined in the Section (I). Given the motion parameters at time t, the translated and scaled template region of the current frame Y_t (called as ROI, region of interest) can be computed using [14, 6]:

$$ROI(U_t) = round\Big(\mathbf{J}U_t + \begin{bmatrix} i_0 \\ j_0 \end{bmatrix}\Big) \tag{4.4}$$

$$\text{where, } \mathbf{J} = \begin{bmatrix} [(i_0 - \tilde{i}_0) & \mathbf{1} & \mathbf{0}] \\ [(j_0 - \tilde{j}_0) & \mathbf{0} & \mathbf{1}] \end{bmatrix}$$

i_0 and j_0 are M-dimensional vectors where M denotes the number of pixels in I_0. $\tilde{i}_0 = \text{mean}(i_0)$ and $\tilde{j}_0 = \text{mean}(j_0)$. Thus the observation model is similar to the one given in [14, 6]

$$Y_t(ROI(U_t)) = \Phi\Lambda_t + I_0 + Z_t \tag{4.5}$$

Comparing with the system model given in (3.3), we can say $\tilde{h}(Y_t, X_t) = Y_t(ROI(U_t)) - \Phi\Lambda_t - I_0$. Here Z_t is assumed to be independent and identically distributed (i.i.d.) Gaussian given by $Z_t \sim \mathcal{N}(0, \Sigma_0\ )$, where $\Sigma_0 = \sigma_0^2\mathbf{I}$ and σ_0^2 denotes the variance of individial pixel noise. Thus for our system model, given the state vector $X_t = [U_t,\, T_t,\, \Lambda_t]^\top$, we have the following Observation Likelihood (OL),

$$\begin{aligned} p(Y_t|X_t) &= p(Y_t(ROI(U_t))|\Lambda_t)p(Y_t(ROI(U_t)^c)|\Lambda_t) \\ &\propto exp(-\frac{Y_t(ROI(U_t)) - \Phi\Lambda_t}{2\sigma_o^2})p(Y_t(ROI(U_t)^c)) \end{aligned}$$

ROI^c denotes the pixels outside the ROI and it does not depend on the state vectors U_t or Λ_t. Hence we can write the OL as:

$$\begin{aligned} OL(X_t) &\triangleq p(Y_t|U_t, \Lambda_t, N_t) \propto p(Y_t(ROI(U_t))|\Lambda_t) \\ &= \Pi_{n=1}^{M}[\mathcal{N}([Y_t(ROI(U_t))]; (\Phi_{T_t}(\Lambda_t)_{T_t} + T_0)_n, \sigma_o^2)] \\ &= \Pi_{n=1}^{M}[\mathcal{N}([Y_t(ROI(U_t))]; (\Phi\Lambda_t + T_0)_n, \sigma_o^2)] \\ &= \frac{1}{\sqrt{2\pi}\sigma_0 M} exp(-\frac{Y_t(ROI(U_t)) - \Phi\Lambda_t}{2\sigma_o^2}) \end{aligned} \tag{4.6}$$

where $[\,]_n$ denotes the n^{th} element of a vector. Hence comparing with (3.2), we have $f_Z(\tilde{h}(Y_t, X_t)) = \frac{1}{\sqrt{(2\pi)}\sigma_0} exp[-\frac{\|Y_t(ROI(U_t)) - \Phi\Lambda - I_0\|}{2\sigma_0^2}]$ The pixels outside the ROI is assumed to have intensities that do not depend on U_t and Λ_t. The Conditional likelihood of Λ_t given a realization $U_t^{(i)}$ of U_t, is defined as:

$$CL^{(i)}(\Lambda_t) = OL(\Lambda_t, U_t^{(i)}) \tag{4.7}$$

4.1.2 Support Change in Real Video

To show that there is a slow support change in real-time data, we take consecutive 20 frames from a video and plot the support change with respect to time as shown in Fig. 4.1. For sparse signals, the support is clearly $T = \{i \in [1, n] : (\Lambda_t)_i^2 > 0\}$, but in real data the elements outside the sparse vector may not be exactly 0. In such cases we take $b\% - Energy\, Support$ [27] i.e, at time t, the support T_t is calculated as: $T = \{i \in [1, n] : (\Lambda_t)_i^2 > \alpha\}$, where α is the largest real number for which T contains at least $b\%$ of the signal energy.
The change in support is computed as follows: For time $t > 0$

i. The illuminated face template are handmarked manually for each frame in the training dataset.

ii. The corresponding illumination vector Λ_t is computed from the face template I_t as : $\Lambda_t = (\Phi^\top \Phi)^{-1}\Phi^\top(I_t - I_0)$ using approximations $I_t = \Phi\Lambda_t + I_0$

iii. Compute support $T_t := \left\{ j \in [1, n] : \dfrac{\sum\limits_j x_j^2}{\sum\limits_{i=1}^{n} x_i^2} \geq 0.99 \right\}$
i.e, $b = 99$ in Fig. 4.1 .

iv. The change in support can be either given as any element added to or deleted from the support. We compute additions as: $\frac{|T_t \backslash T_{t-1}|}{|T_{t-1}|}$ and deletions as: $\frac{|T_{t-1} \backslash T_t|}{|T_{t-1}|}$ and the supposrt size is given as $\frac{|T_t|}{D}$

We estimate the support size for two different videos. In Fig. 4.1 (a) and (b) we give the various time instances of the illuminated face template and we see that the illumination is not uniform over the face i.e, spatially varying. (a) correspond to a situation where a person walks under a tree in daylight and (b) correspond to a situation where a person across a window. In Fig. 4.1 (c) and (d) we plot the size of the changes (additions and deletions) of the Legendre support (set of indices of the large Legendre basis coefficients) at each time as a ratio of the support size corresponding to illumination conditions in Fig. 4.1 (a) and (b) respectively. From Fig. 4.1(c), we see here that though the support size remains more or less constant for most of

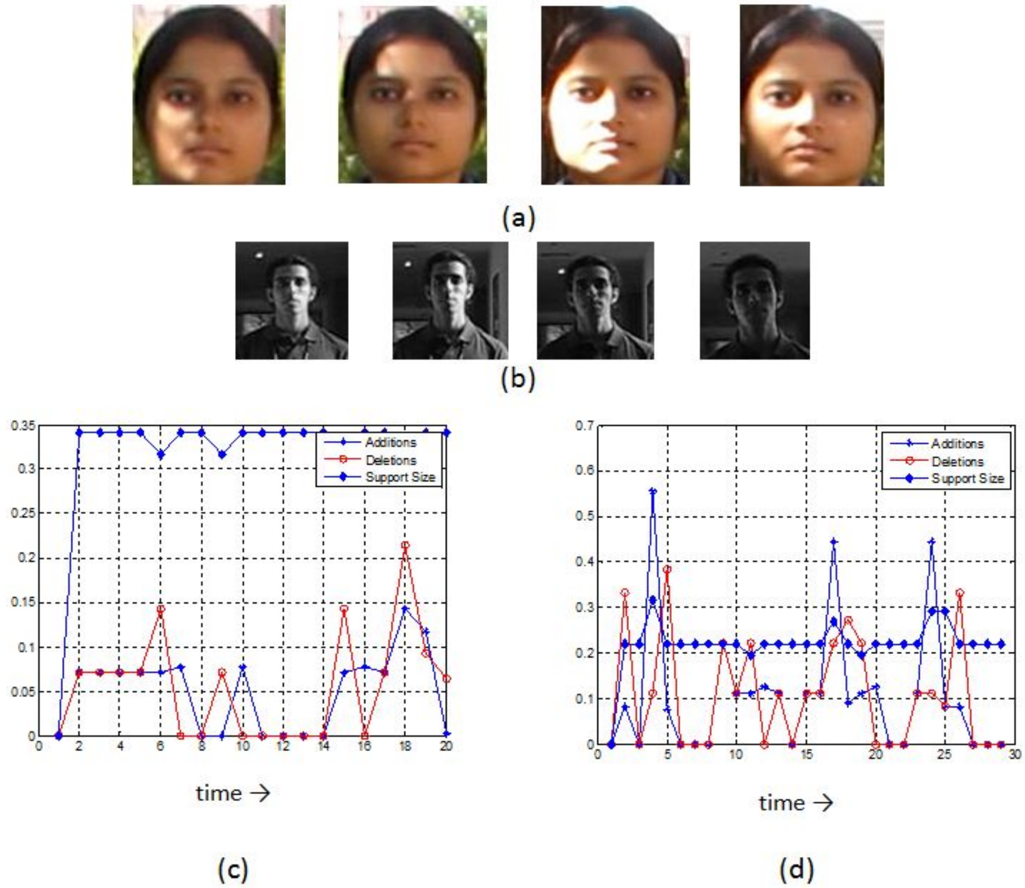

Figure 4.1 Support Change of Illumination Vector in real time video. additions $= \frac{|T_t \setminus T_{t-1}|}{|T_{t-1}|}$ and deletions $= \frac{|T_{t-1} \setminus T_t|}{|T_{t-1}|}$ and the normalised support size is given as $\frac{|T_t|}{D}$

the time period and changes only at after a few intervals. Also notice that there the additions and deletions from the support do not occur at every time slot, but, at all times, the changes are less than 10% of the actual support size. But in case of Fig. 4.1(d), we see that the change in support is much large and more frequent but the support size here also remains more or less constant for most of frames. So we can say that illumination vector is sparse in the Legendre basis and the sparsity pattern changes slowly over time.

4.2 Pafimocs Algorithms

Considering the system models given above the PaFiMoCS algorithms (2), (3) for this problem is given by algorithms (4) and (5) respectively.

Algorithm 4 PafiMoCS : Estimation of sparse vector Λ_t from observation Y_t and importance sampling on U_t, and state vector $X_t = [U_t, \Lambda_t]$

1. At $t = 0$, U_0, T_0 and Λ_0 are known,
   For $i = 1....N_{pf}$; assign particle set as:

   $U_0^{(i)} \sim \mathcal{N}(U_0, \Sigma_{U_0})$
   $T_0^{(i)} = T_0$
   $(\Lambda^{(i)})_{T_0} \sim \mathcal{N}(\Lambda_{T_0}, \Sigma_{\Lambda_0})$
   $(\Lambda^{(i)})_{T_0^c} = \mathbf{0}$

2 At each time $t > 0$ and for $i = 1....N_{pf}$:

   a. Importance Sample $U_t^{(i)} \sim \mathcal{N}(U_{t-1}^{(i)}, \Sigma_u)$

   b. Compute $ROI(U_t^{(i)})$, using (4.4)

   c. Using current observation Y_t compute $Y_t(ROI(U_t^{(i)}))$

   d. Perform Reg-Mod-CS:

$$\begin{aligned}\Lambda_t^{(i)} &= \arg\min_{\Lambda} [\gamma\|(\Lambda_{T^c})\|_1 + \frac{\|Y_t(ROI(U_t^{(i)})) - \Phi\Lambda - I_0\|_2^2}{2\sigma_0^2} \\ &+ \frac{\|(\Lambda - \Lambda_{t-1}^{(i)})_T\|_2^2]}{2\sigma_l^2}\end{aligned}$$

   where $T = N_{t-1}^{(i)}$.

   e. Compute $T_t^{(i)} = \{j; |(\Lambda_t^{(i)})|_j > \alpha\}$

   f. Assign Importance weight as:

$$\omega_t^{(i)} \propto OL(U_t^{(i)}, \Lambda_t^{(i)})STP(\Lambda_t^{(i)}; \Lambda_{t-1}^{(i)}, T_t^{(i)})STP(T_t^{(i)}; T_{t-1}^{(i)})$$

   g. Resample and reset weights. Increment t and go to step 1.

The block diagram for algorithm (4) and algorithm (5) are given in fig. (4.2) and fig. (4.3) respectively.

Algorithm 5 PafiMoCS : Estimation of sparse vector Λ_t from observation Y_t and importance sampling on U_t and N_t and state vector $X_t = [U_t, T_t, \Lambda_t]$

1. At $t = 0$, U_0, T_0 and Λ_0 are known,
   For $i = 1....n_{pf}$; assign particle set as:

   $$U_0^{(i)} \sim \mathcal{N}(U_0, \Sigma_{U_0})$$
   $$T_0^{(i)} = T_0$$
   $$(\Lambda^{(i)})_{T_0} \sim \mathcal{N}(\Lambda_{T_0}, \Sigma_{\Lambda_0})$$
   $$(\Lambda^{(i)})_{T_0^c} = 0$$

2 At each time $t > 0$ and for $i = 1....n_{pf}$:

a. Importance Sample on signal support :

$A_t^{(i)} \sim \text{Ber}(T_{t-1}^{(i)\,c}, p_{add})$ and
$R_t^{(i)} \sim \text{Ber}(T_{t-1}^{(i)}, p_{rem})$.

Get $T_t^{(i)} = (T_{t-1}^{(i)} \cup A_t^{(i)}) \setminus R_t^{(i)}$

b. Importance Sample $U_t^{(i)} \sim \mathcal{N}(U_{t-1}^{(i)}, \Sigma_u)$

c. Compute $ROI(U_t^{(i)})$, using (6)

d. Using current observation Y_t compute $Y_t(ROI(U_t^{(i)}))$

e. Perform Reg-Mod-CS:

$$\Lambda_t^{(i)} = \arg\min_{\Lambda} [\gamma \|(\Lambda_{T^c})\|_1 + \frac{\|Y_t(ROI(U_t^{(i)})) - \Phi\Lambda - I_0\|_2^2}{2\sigma_0^2} + \frac{\|(\Lambda - \Lambda_{t-1}^{(i)})_T\|_2^2]}{2\sigma_l^2}$$

where $\hat{T} = T_t^{(i)}$.

f. Compute $T_t^{(i)} = \{j;\ |(\Lambda_t^{(i)})|_j > \delta\}$

g. Assign Importance weight as:

$$\omega_t^{(i)} \propto OL(U_t^{(i)}, \Lambda_t^{(i)}) STP((\Lambda_t^{(i)}), T_t^i, (\Lambda_{t-1}^{(i)}), T_{t-1}^{(i)})$$
$$\propto OL(U_t^{(i)}, \Lambda_t^i) STP(\Lambda_t^{(i)}; \Lambda_{t-1}^{(i)}, T_t^{(i)})$$

h. Resample and reset weights. Increment t and go to step 1.

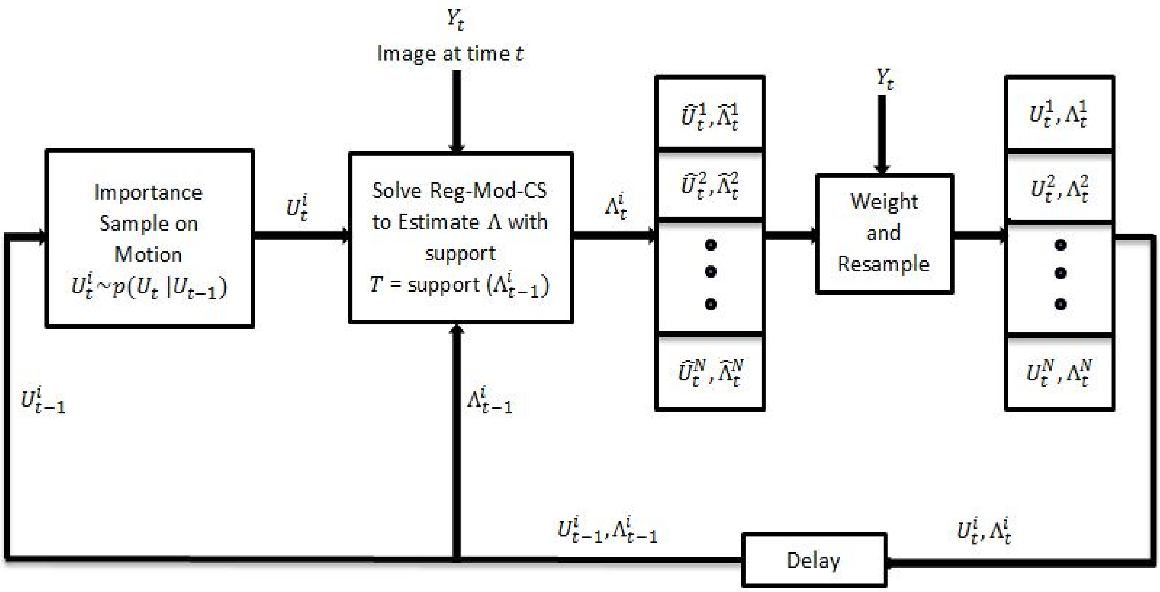

Figure 4.2 Block Diagram for Algorithm (3).

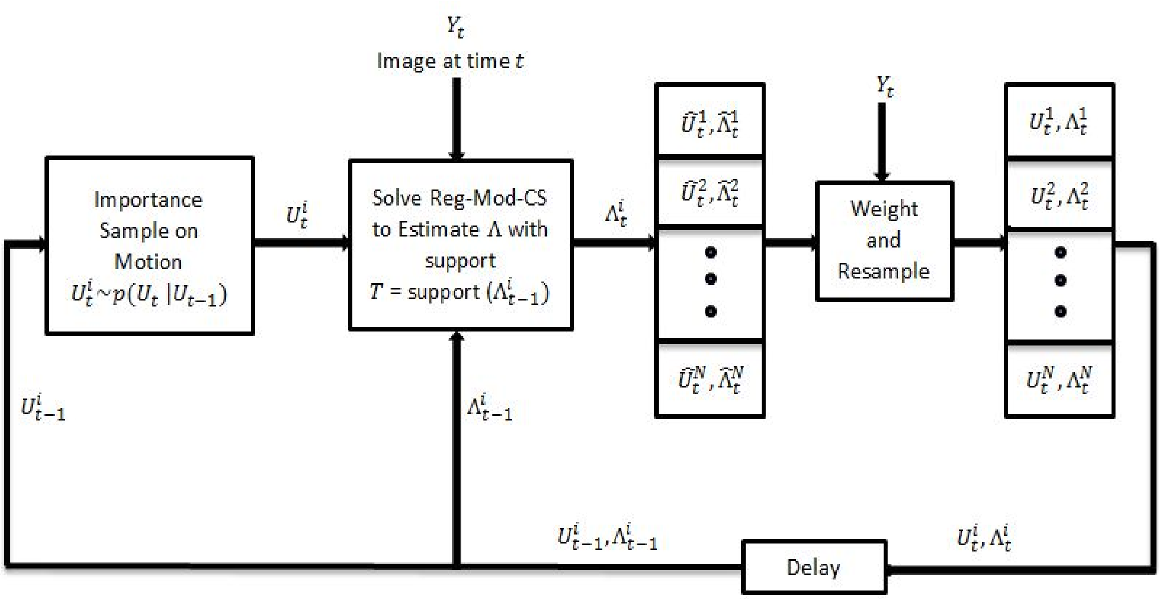

Figure 4.3 Block Diagram for Algorithm (4).

CHAPTER 5. EXPERIMENTAL RESULTS

5.0.1 Simulation

A video is simulated using a random walk model to a persons face on an arbitrary background and a Legendre sparse illumination model is applied to the face template, such that the support of the Legendre basis changes every few frames also based on a random walk model. We use Monte-Carlo to estimate the normalised-mean squared error for Λ. The following steps are implemented in simulating the video

1. Initialization: $T_0 \sim$ Choose any random 5 numbers from 1 to D (here $D = 41$),and hence $|T_0| = 5$, which is the dimension of the support at time $t = 0$. Hence the support size is around 10% of the illumination vector. Initial motion parameters is set as $U_0 = [0,\ 0,\ 0]^\top$ and $(\Lambda_0)_{T_0} = \mathbf{1}$ and $(\Lambda_0)_{T_0^c} = \mathbf{0}$.

2. for $t > 0$

   (a) The illumination model is changed every 5 frames. A sparse vector of the legendre basis coefficients are chosen in the following way:

   $A_t \sim \mathrm{Ber}((T_{t-1})^c, p_a)$;

   $R_t \sim \mathrm{Ber}((T_{t-1}), p_r)$;

   $T_t \sim ((T_{t-1}) \cup A_t) \backslash R_t$;

   compute $(\Lambda_t)_{T_t} \sim \mathcal{N}((\Lambda_{t-1})_{T_t}, \sigma_l^2 \mathbf{I})$ and $(\Lambda_t)_{T_t^c} = 0$.

   We consider p_a and p_r such the change in the support size is small and the support size remains nearly constant. So we choose $p_a = 0.06$, a small number which would ensure that a small number of coefficients are added to the support and p_r is chosen to be 0.7, using the fact $(D - k)P_a = kp_r$, where $k = |T_t|$, such that the support size k would remain more or less constant. We fix $\sigma_l^2 = 10^{-2}$, which implies the slow sparsity change over time.

(b) The illumination of the frame at t is then simulated as : Compute the legendre basis vector $\mathbf{P}$ and compute Φ as given by (4.2). I_0 is the initial template (person's face) at time $t = 0$.

(c) The face template at time t is given as $I_t = \Phi\Lambda_t + I_0 + \Sigma_0$, where $\Sigma_0 = \sigma_0^2\mathbf{I}$.

(d) A random walk model is applied to the object shape U_t and we generate the motion vector $U_t \sim \mathcal{N}[U_{t-1}, \Sigma_U]$ such that,

$s_t \sim \mathcal{N}(s_{t-1}, \sigma_s^2)$;

$u_t^x \sim \mathcal{N}(u_{t-1}^x, \sigma_{u_x}^2)$ and

$u_t^y \sim \mathcal{N}(u_{t-1}^y, \sigma_{u_y}^2)$

We take $\sigma_s^2 = 0.0001$, $\sigma_{u_x}^2 = 0.2$ and $\sigma_{u_y}^2$=0.001

(e) Compute $(ROI(U_t))$ using (4.4) and reshape the face template and compute $Y_t(ROI(U_t))$ $= I_t$, where Y is the image frame (the fixed background). The recomputed Y_t is the video frame at time t.

3. Here we choose the threshold α at each step so that it would ensure the support to contain 99% of the signal energy. The normalised mean squared error, (NMSE) = $||\hat{\Lambda}_t - \Lambda_t||_2^2/||\Lambda_t||_2^2$, using 100 particles for $\sigma_0^2 = 10^{-6}$.

4. The result for the normalised error of Λ is compared with PF-MT, PF, Aux-PF.

5.0.2 Video Sequence

Here we compare our proposed algorithms with other existing PF algorithms and show that the two Pafimocs algorithms outperforms the rest. We consider llumination conditions which can account for high frequency spatial variation of light when a person moves under a tree, a region of frequent shades and light, through an illuminated corridor and move towards or away from a light source.

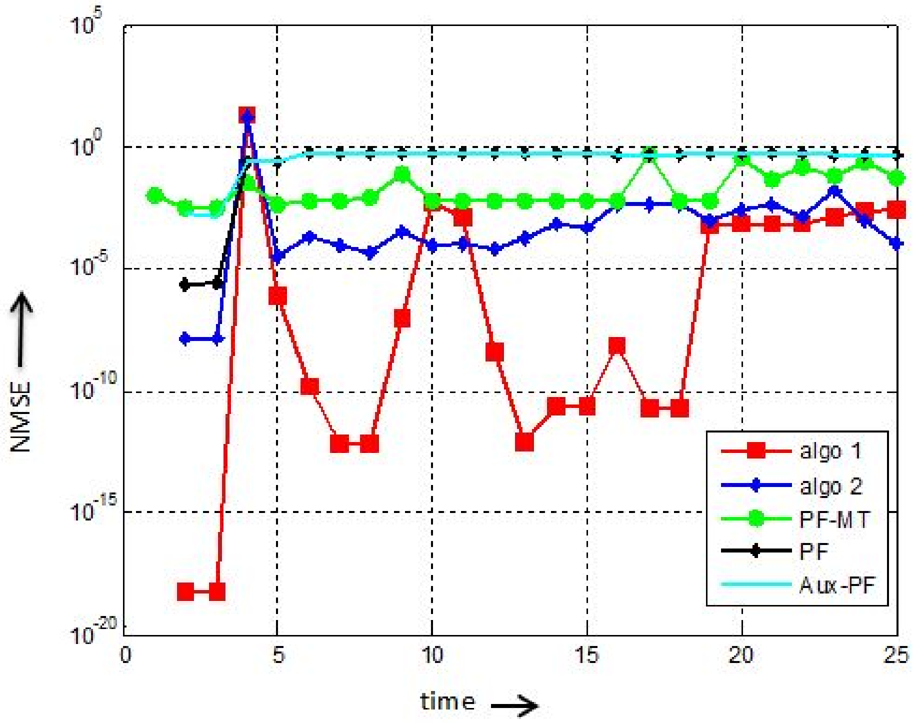

Figure 5.1 Normalised mean squared error for Λ for various PF-based methods, where NMSE $= \frac{\|\hat{\Lambda}_t - \Lambda_t\|_2^2}{\|\Lambda_t\|_2^2}$

In the first experiment shown in fig. (5.3), we use a training sequence. Here we hand mark the centroids of the face to get the location of the face in each frame to learn the motion parameters. Also the corresponding values of Λ for the first 20 frames are obtained as given in Section $IV - B$. The covariance matrices of the change of U_t and of Λ_t, σ_Λ and σ_u are estimated using standard maximum likelihood estimation applied to $(U_t - U_{t-1})$ and $(\Lambda_t - \Lambda_{t-1})$. For all the PF algorithms, we used a fixed particle size of $N_{pf} = 60$. The tracking performance of PAFIMOCS in the presence of illumination change was compared with several other PF-based algorithms like - PF-MT [6] (both using $D = 7$ and $D = 41$), Auxiliary-PF [20] and PF-Gordon [12]. PF-Doucet [10] cannot be implemented here for reasons similar to that given in [6]. PF-MT with both 7 and 41 dimensions fail here. That is because when we use Λ as a 7-dimensional vector, it does not include the higher order Legender basis coefficients, that contribute mostly to the complex illumination of the face as studied in Section $IV - B$. In our case as we see from fig. 4.1, the illumination vector is sparse and the support size changes

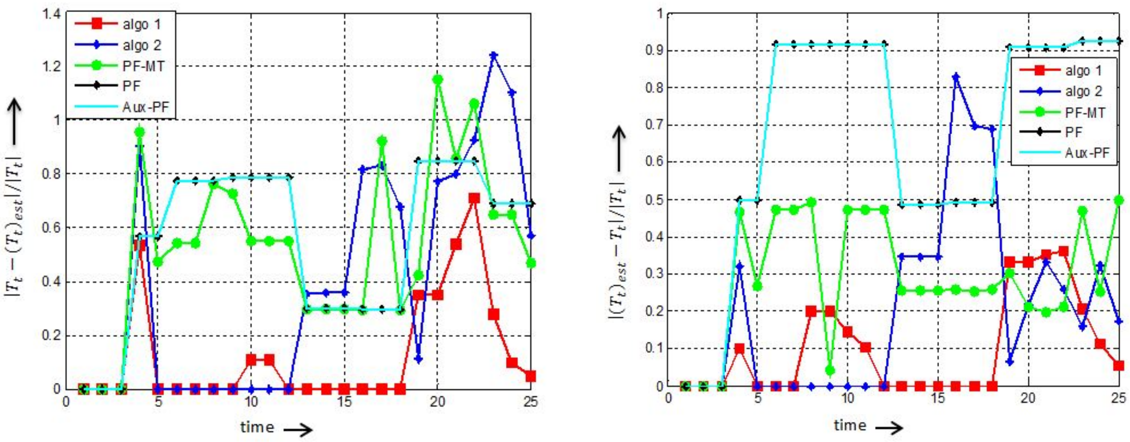

Figure 5.2 Normalised support error estimation of Λ plot for various PF-based methods.

slowly over time i.e, only certain directions of the basis vector contribute to the illumination. PF-MT can detect a slow change in the vector in every direction of the basis. Hence when we use use PF-MT with 41 dimension, it fails to detect the sparsity of a vector. So it looses track after a certain time step, before which the illumination is more or less constant over the face i,e with much less spatial variation. Aux-PF fails from a much earlier track compared to the other algorithms and PF-Gordon also fails as for a 44 - dimensional state vector just 60 particles are insufficient to detect the true state vectors.

For our second experiment fig. (5.4), we show when a person walks through a corridor across a window. We show the comparisons for frames 16, 23 and 30. Here we use $N_pf = 120$. But still we see that PF-MT using both 7 and 41 dimensional illumination vector fails to track the target. Since PF-MT also fails we can say that both PF-Gordon and Auxilliary PF are likely to fail with 120 particles. For both our experimnets we do not compare with PF-Gordon and Auxilliary PF using 7-dimensional illumination vector as in [6], it has been already shown that PF-Gordon and Auxilliary-PF both fails to track with 7-dimensional illumination vector using 100 particles as both these algorithms are not able to detect the illumination vector correctly.

In fig (5.5), we show more complex immulinaiton conditions where the illumination sources are corridor light and at certain intervals light coming through doors of a room of maybe sunlight through window. We show that both our algorithms are able to track in such complex illumination conditions for nearly frames. In fig (5.7), we show an experiment when do an experiment where a person moves through shaded regions. and in fig (5.6) we show the tracking of a person moving through a well lit subway station obtained from *https://redpill.ecn.purdue.edu/ hvact/*

.

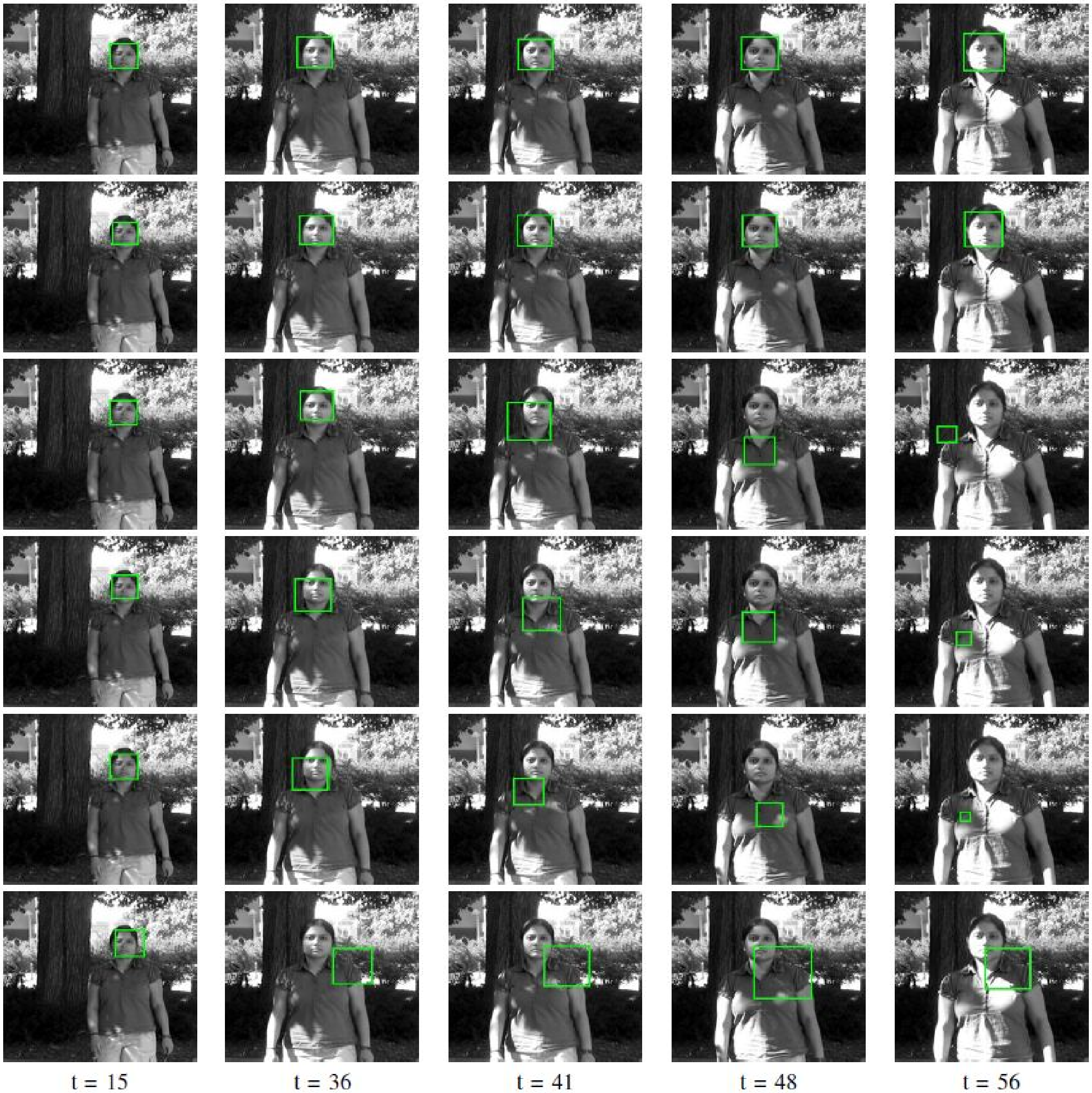

Figure 5.3 Visual comparison of various methods for face tracking across illumination changes, when a person moves under a tree. Here we show comparisons of our algorithm with other existing PF algorithms. The first row correspond to PAFIMOCS-algorithm 1 and the second row correspond to PAFIMOCS-algorithm 2. the thurd and fourth row correspond to PF-MT with illumination vector 7 and 41 dimensions respectively. The fifth row correspond to PF-Gordon and the last row correspond to Auxilliary-PF. We show the comparison for frames 15, 36, 41, 48 and 56 respectively.

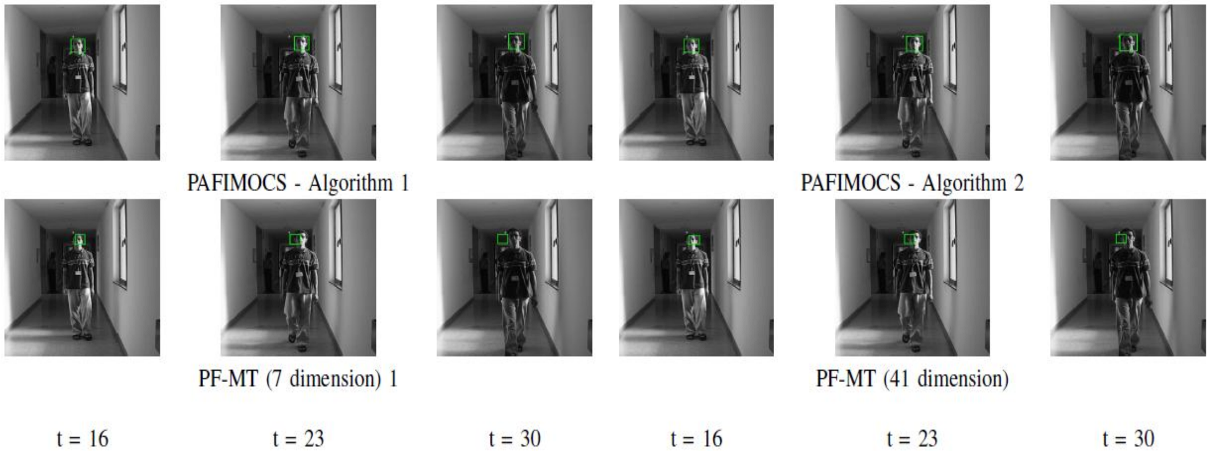

Figure 5.4 The first 3 figures of the first row correspond to our Algorithm 1 and the last 3 figures correspond to Algorithm 2. The second row first 3 figures correspond to PF-MT with 7 dimensional illumination vector and the last 3 figures correspond to PF-MT with 41 dimensional illumination vector.

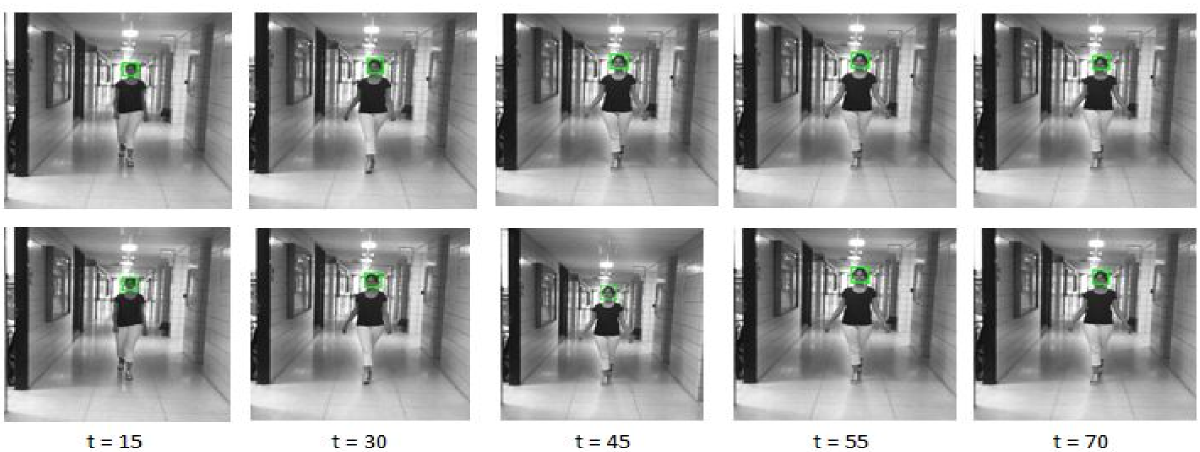

Figure 5.5 The figures show tracking when a person walking through a corridoor. The first row correspond to algorithm 1 and the second row correspond to algorithm 2

Figure 5.6 The figures show tracking when a person walking in a subway station. The first row correspond to algorithm 1 and the second row correspond to algorithm 2

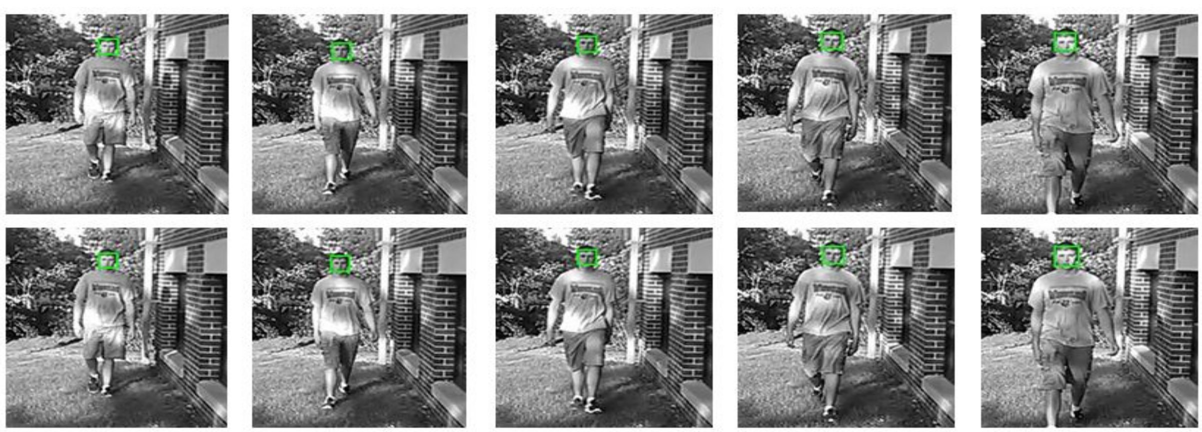

Figure 5.7 The figures show tracking when a person walking in any shaded region.The first row correspond to algorithm 1 and the second row correspond to algorithm 2

CHAPTER 6. Conclusion and Future Work

We have proposed an algorithm for sequential estimation (i.e. tracking) of sparse signals from a small number of linear measurements. The algorithm like PF-MT divides the state space into smaller dimensional effective basis and a large dimensional residual space, which is sparse. It utilizes a dynamic prior model on both sparsity pattern change as well as on signal dynamics on the known part of the support and the smaller dimensional effective basis. It can be consedered as a merger between particle filtering and compressive sensing and hence the name - Particle Filtered Modified Compressive Sensing(PaFiMoCS). Our simulation experiments prove PaFiMoCS to be more promising as compared to other PF related algorithms. However, the proposed illumination PaFiMoCS algorithm can very easily be adapted to other representations of the target e.g. feature based approaches. A similar approach can be also be developed to jointly handle appearance change due to illumination as well as other factors like 3D pose change, by using the more sophisticated models of recent work. Similarly, illumination can also be represented using other parameterizations or using different basis vectors. Also the same algorithm can be developed such that it is robust to occlusions where the occlusion model can itself be designed as a sparse vector or as has been designed in [6]. Another application that can be handled with this is contour tracking where the motion is small dimensional and the deformation in shape could be large dimensional and designed as Fourier sparse vector.

BIBLIOGRAPHY

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *TSP*, 50(2):174–188, February 2002.

[2] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IT*, 52(2):489–509, February 2006.

[3] E. Candes and T. Tao. Decoding by linear programming. *IT*, 51(12):4203 – 4215, Dec. 2005.

[4] R. Chen and J.S. Liu. Mixture kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508, 2000.

[5] Scott Shaobing Chen, David L. Donoho, Michael, and A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[6] S. Das, A. Kale, and N. Vaswani. Particle filter with mode tracker (pf-mt) for visual tracking across illumination changes. *TIP*, April 2012.

[7] Samarjit Das. Particle filteling on large dimensional state space and applications in computer vision. *Phd Thesis 2010 Iowa State University.*

[8] Samarjit Das and N. Vaswani. Efficient importance sampling techniques for large dimensional and multimodal posterior computations. In *IEEE Digital Signal Processing/SPE Workshop*, 2009.

[9] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, April 2006.

[10] A. Doucet. On sequential monte carlo sampling methods for bayesian filtering. In *Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering*, 1998.

[11] A. Doucet, N. deFreitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/nongaussian bayesian state estimation. *IEE Proceedings-F (Radar and Signal Processing)*, pages 140(2):107–113, 1993.

[13] M. Isard and A. Blake. Condensation: Conditional Density Propagation for Visual Tracking. *IJCV*, pages 5–28, 1998.

[14] A. Kale and C. Jaynes. A joint illumination and shape model for visual tracking. In *CVPR*, pages 602–609, 2006.

[15] J. H. Kotecha and P. M. Djuric. Gaussian particle filtering. *TSP*, pages 2592–2601, Oct 2003.

[16] J. H. Kotecha and P. M. Djuric. Gaussian sum particle filtering. *TSP*, pages 2602–2612, Oct 2003.

[17] Wei Lu and N. Vaswani. Regularized modified bpdn for noisy sparse reconstruction with partial erroneous support and signal value knowledge. *IEEE Trans. in Signal Proc.*, 2012.

[18] Wei Lu and Namrata Vaswani. Modified basis pursuit denoising (modified-bpdn) for noisy compressive sensing with partially known support. In *IEEE Intl. Conf. Acous. Speech. Sig.Proc.(ICASSP)*, 2010.

[19] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 1995.

[20] M. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *J. Amer. Stat. Assoc*, 94, 1999.

[21] R. Sarkar, S. Das, and N. Vaswani. Pafimocs: Particle filtered modified-cs and applications in visual tracking across illumination change. *TIP*, page (under review).

[22] T. Schn, F. Gustafsson, and P. Nordlund. Marginalized particle filters for nonlinear state-space models. *TSP*, 2005.

[23] N. Vaswani. Particle filters for infinite (or large) dimensional state spaces-part 2. In *ICASSP*, 2006.

[24] N. Vaswani. Pf-eis & pf-mt: New particle filtering algorithms for multimodal observation likelihoods and large dimensional state spaces. In *ICASSP*, 2007.

[25] N. Vaswani. Particle filtering for large dimensional state spaces with multimodal observation likelihoods. *TSP*, pages 4583–4597, October 2008.

[26] N. Vaswani and S. Das. Particle filter with efficient importance sampling and mode tracking (pf-eis-mt) and its application to landmark shape tracking. In *Asilomar Conf. on Sig. Sys. Comp.*, 2007.

[27] N. Vaswani and W. Lu. Modified-cs: Modifying compressive sensing for problems with partially known support. *IEEE Trans. Signal Processing*, September 2010.

[28] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001.

[29] Greg Welch and Gray Bishop. An introduction to Kalman Filters. *SIGGRAPH*, 2001.