"Shape Activities" : Dynamic Stochastic Models for Moving/Deforming Shapes with Application to Abnormal Activity Detection *

Namrata Vaswani[a,b] Amit Roy Chowdhury[b], Rama Chellappa[a,b]

[a]Dept. of Electrical and Computer Engineering,

[b]Center for Automation Research,

University of Maryland, College Park, MD 20742, USA

{namrata,amitrc,rama}@cfar.umd.edu

Abstract

The changing configuration of a group of moving landmarks can be modeled as a moving/deforming shape. The landmarks could be moving objects (people/vehicles/robots) or rigid parts of an articulated shape like the human body. The deformation of a moving/deforming shape can be split into rigid motion of an average shape and its non-rigid deformations. We use the term **"shape activity"** *to denote a particular stochastic model for shape deformation. We propose dynamical models for non-rigid shape deformation and for motion. Noise in the observed landmark locations makes the system partially observed and the mapping between observation and shape space makes it non-linear. The partially observed shape activity model is used to represent an activity performed by a group of moving objects. An abnormal activity is then defined as a change in the shape activity model with change parameters being unknown. We propose a change detection strategy using particle filters which can detect both slow and drastic changes in partially observed nonlinear systems. Results are shown on a real abnormal activity detection problem involving multiple moving objects (treated as point objects).*

1 Introduction

In this work, we develop models for the configuration dynamics of a group of moving landmarks (here point objects) in shape space. Shape of a group of discrete points (known as 'landmarks') is defined by Kendall [1] as all the geometric information that remains when location, scale and rotational effects are filtered out. Statistical

* Partially supported by the DARPA/ONR Grant N00014-02-1-0809

shape theory began in the late 1970s and has evolved into viable statistical approaches for modeling the shape of an object with applications in object recogniton and matching. For a dataset of similar shapes, the shape variability can be modeled in the tangent plane to the shape space. The tangent plane is a vector space and hence techniques from linear multivariate statistics can be used to model shape variability in tangent space.

In our work, we model (statistically) the dynamics of a time sequence of deforming and moving shapes in the tangent space to shape space. To model the configuration dynamics, we also define motion models (models for translation, isotropic scaling and rotation). We use the term **"shape activity"** to denote a particular stochastic model for shape deformation and define a "stationary shape activity" as one in which the mean shape remains constant and the shape deformation process is stationary. We use these ideas to model real activities performed by a group of moving and interacting objects. One example of a stationary shape activity, that we study in this paper, is that of people (treated as point objects) deplaning and moving towards the terminal at an airport (figure 2(a)). Our framework can be used to model normal activity and detect abnormal activity as a deviation from the normalcy model. We are able to detect both spatial and temporal abormalities (terminology borrowed from [2]). It is interesting to note that the "shape activity" framework can also be used to model the dynamics of articulated shapes like the human body (the different rigid parts of the human body forming the landmarks) and thus represent different actions.

Given a training sequence of configurations from a "stationary shape activity", we estimate the mean shape after normalizing for translation, scale and rotation as described in [3]. For the sake of completeness, we first present some definitions and methods for shape analysis in section 2. The shape dynamics is modeled by a linear Gauss Markov model in the tangent space to shape space with the mean shape as the pole. We discuss the model and its training in section 3.1. A test sequence of observations (obtained using an automated object detection algorithm) would have significant observation noise associated with it, thus making the state of the system partially observed. In section 3.2, we define a partially observed dynamical model [4] for a "shape activity", with shape and motion (rotation, scale, translation) forming the state and the observed landmark locations forming the observation vector. We also describe how to model "non-stationary shape activities" in section 3.3. Since the partially observed model is nonlinear, we use a particle filter (PF) [5] to estimate (filter out) the posterior probability distribution of shape from the noisy observations. We describe particle filtering in section 3.4 and discuss the advantage of using a PF over an Extended Kalman Filter (EKF) [6], in 3.4.1. In section 4, the problem of detecting abnormal activities is posed as a change detection problem and a strategy for detecting both slow and drastic changes is proposed.

In section 5.1, we discuss how the PF can also be used for "tracking observations " as in the CONDENSATION algorithm [7] to improve the measurement algorithm for obtaining observations [1] . We also describe how our framework can be adapted for classifying between shape activities (section 5.2.1) and for tracking a sequence of shape activities (section 5.2.2). Finally, possible applications of our framework are discussed in section 6, experimental and simulation results presented in section 7 and conclusions are given in section 8.

1.1 Related Work

Some of the commonly used representations for shape are Fourier descriptors [8], splines [9] and deformable snakes all of which model the shape of continuous curves. But in our work we are attempting to model the dynamics of a group of discrete landmarks (which could be moving point objects or moving parts of an articulated object like the human body). Since the data is inherently finite dimensional, using infinite dimensional representations of a continuous curve is not necessary and hence we look only at the representation of shape in $\Re^n$ (modulo Euclidean similarity transformations) which was first defined by Kendall in 1977. Active Shape Models started by Cootes et al [10] also consider the shape of points in $\Re^n$. In [11], they define 'Point Distribution Models' which are principal component models for shape variation using Procrustes residuals.

There has been a lot of work in defining probability distributions in (Kendall's) shape and preshape space and also in analyzing datasets of similar shapes in the tangent space at the mean (discussed in chapter 6, 7, 11 of [3], [12], [13] and references therein). Many models for shape deformation of one shape into another have been proposed which include affine deformation, thin plate splines, and principal and partial warp deformations (discussed in chapter 10 of [3]). But none of these define dynamical models for time seqeunces of shapes. We propose in this paper, partially observed dynamical models for stationary and non-stationary shape activities. Our model for non-stationary shape activities is similar in spirit to those in [14] and [15] where the authors define dynamical models for motion on Lie groups and Grassmann manifolds, respectively, using piecewise geodesic priors and track them using particle filtering.

[1]We clarify the difference between filtering, "tracking" and "tracking observations" since these terms are used very often in the paper. **Filtering** is estimating the actual process or state X_t from noisy observations upto the current time, $Y_{0:t}$. System model based filtering algorithms like Kalman filter or particle filter perform filtering as a two step process of prediction (predicting X_t and Y_t using $Y_{0:t-1}$) and update (updating the prediction of X_t using the observed value of Y_t) and this is also known as **tracking**. The purpose of tracking can be either just filtering, or the predicted observation can be used in conjunction with the measurement algorithm to **"track observations"**, i.e. to search for Y_t in the vicinity of its prediction. The term "tracking" is quite often used in literature in the sense of "tracking observations". Also note that, the term "tracks" denotes the predicted observation, $\hat{Y}_t$ and "tracking error" denotes prediction error, $||Y_t - \hat{Y}_t||^2$.

There is a huge body of work in computer vision on modeling and recognition of activity, human actions and events. The work can be classified (based on the formalisms used) as Bayesian networks (BNs) and Dynamic Bayesian networks (DBNs) [16, 17]; Hidden Markov models for representing activity [18, 19] ; stochastic grammars [20]; and factorization method based approaches [21, 22]. In [2], the authors perform clustering to learn the co-occurrence statistics of individual objects and their interactions with other objects. [23] is another work which treats events as long spatio-temoral objects and clusters them based on their behavioral content. In [24], action "objects" are represented using generalized cylinders with time forming the cylinder axis. Now, [2, 21, 22, 23, 24] are non-parametric approaches to activity/event recognition, while HMMs, stochastic grammars, BNs and DBNs are model based approaches. Our work also defines a parametric model for activity performed by a group of objects but there are two main differences. First, we treat objects as point objects and hence we can get our observations from low resolution video or even from other sensors like radar, acoustic or infra-red (our approach is sensor independent). Second, we provide a compact and global framework for modeling the interactions and independent motion of multiple moving objects by treating them as a deformable shape. Also, using shape to represent group activity or human actions makes the activity representation invariant to translation/zoom/ in-plane rotation of a camera or any other sensor (under scaled orthography).

Particle filters [5] have been used extensively in computer vision for tracking [2] moving objects [4, 7, 25]. [26] uses particle filtering to track multiple moving objects but it uses separate state vectors for each object and defines data association events to associate the state and observation vectors. In our work, we represent the combined state of all moving objects using the shape of its configuration and define a dynamic model for it. We track using a PF to filter out the shape from noisy observations of the object locations and use it for abnormal activity detection. We define an abnormal activity as a change which could be slow or drastic and whose parameters are unknown. Change detection in linear systems has been studied extensively. Linearizing the nonlinear system and using linear change detection algorithms has been the main strategy to deal with nonlinear systems (see [27] for details). Change detection in nonlinear systems using particle filters is discussed in [28]. But it assumes that the changed system's parameters are known and it deals only with sudden changes. In this paper we propose a statistic for detecting slow changes, with unknown parameters, in nonlinear partially observed systems.

[2] "tracking" here refers to "tracking observations"

2 Shape Preliminaries and Notation

We use "$\arg$" to denote the angle of a complex scalar as well as in "$\arg\min$" for the argument minimizing a function, but the meaning is clear from the context. * is used to denote conjugate transpose. $||.||$ is used for the Euclidean norm of a complex or real vector and $|.|$ for the absolute value of a complex scalar. I_k denotes the $k \times k$ identity matrix and 1_k denotes a k dimensional vector of ones. Also note that to simplify notation we do not distinguish between a random process and its realization. We briefly review below the definitions and tools for statistical shape analysis as described in [3].

Definition 1 *The* **configuration** *is the set of landmarks (which in our case the set of point object locations).* **Configuration matrix** *is the $k \times m$ matrix of Cartesian coordinates of the k ladnmarks in m dimensions. For 2D data ($m = 2$), a more compact representation is a k dimensional complex vector with x and y coordinates forming the real and imaginary parts.* **Configuration space** *is the space of all possible landmarks i.e. $\Re^{km}$.*

Translation Normalization: The complex vector of the configuration (Y_{raw}) can be centered by subtracting out the centroid of the vector, i.e.

$$Y = CY_{raw} \ \text{ where } \ C = I_k - \frac{1_k {1_k}^T}{k}. \tag{1}$$

Definition 2 *The* **pre-shape** *of a configuration matrix (or complex vector), Y_{raw}, is all the geometric information about Y_{raw} that is invariant under location and isotropic scaling. The* **pre-shape space**, *S_m^k, is the space of all possible pre-shapes. S_m^k is a hyper-sphere of unit radius in $\Re^{(k-1)m}$ and hence its dimension is $(k-1)m-1$ (a unit hyper-sphere in $\Re^P$ has dimension $P-1$).*

Scale Normalization: The pre-shape is obtained by normalizing the centered configuration, Y, by its Euclidean norm, $s(Y) = ||Y||$ (known as **scale or size** of the configuration), i.e. $w(Y) = Y/s(Y)$.

Definition 3 *The* **shape** *of a configuration matrix (or complex vector), Y_{raw}, is all the geometric information about Y_{raw} that is invariant under location, isotropic scaling and rotation (Euclidean similarity transformations) i.e. $[z] = \{sY_{raw}R + 1_k\alpha^T : s \in \Re^+, R \in SO(m), \alpha \in \Re^m\}$. The* **shape space** *is the set of all possible shapes. Formally, the shape space, Σ_m^k, is the orbit space of the non-coincident k point set configurations in $\Re^m$ under the action of Euclidean similarity transformations. The dimension of shape space is $M = (k-1)m-1-m(m-1)/2$. It is easy to see that $\Sigma_m^k = S_m^k/SO(m)$, i.e. Σ_m^k is the quotient space of S_m^k under the action of the special orthogonal group of rotations, $SO(m)$.*

Rotation Normalization: Shape, z, is obtained from a pre-shape, w (of centered configuration Y), by rotating it in order to align it to a reference pre-shape γ. The optimal rotation angle is given by $\theta(Y, \gamma) = -\arg(w^*\gamma) = -\arg(Y^*\gamma)$, and the shape, $z(Y, \gamma) = we^{j\theta(Y,\gamma)} = \frac{Y}{s(Y)}e^{j\theta(Y,\gamma)}$.

In this work we deal with $m = 2$ dimensional shapes and hence the configuration vector is represented as a k dimensional complex vector and the shape space dimension is $(2k - 4)$.

Distance between shapes: A concept of distance between shapes is required to fully define the non-Euclidean shape metric space. We use the Procrustes distance which is defined below.

Definition 4 *The* **full Procrustes fit** *of w onto y is [3]*

$$\begin{aligned} w^P(y) &= \hat{\beta}e^{j\hat{\theta}}w + \hat{a} + j\hat{b} \quad \textit{where} \\ \hat{\beta}, \hat{\theta}, \hat{a}, \hat{b} &= \arg\min_{(\beta,\theta,a,b)} D(y, w), \quad D(y, w) = ||y - (\beta e^{i\theta}w + a + jb)||. \end{aligned} \tag{2}$$

If y and w are centered preshapes, it is easy to see that the matching parameters are (result 3.1 of [3])

$$\hat{a} + j\hat{b} = 0, \quad \hat{\theta} = \arg(w^*y), \quad \hat{\beta} = |w^*y| = (y^*ww^*y)^{1/2} \tag{3}$$

Definition 5 *The* **full Procrustes distance** *[3] between preshapes w and y is the Euclidean distance between the Procrustes fit of w onto y, i.e.*

$$D_F(w, y) = \inf_{\beta,\theta,a,b} D(y, w) = ||y - w^P(y)|| = \sqrt{1 - y^*ww^*y} \tag{4}$$

Definition 6 *The* **full Procrustes estimate of mean shape** *(commonly referred to as* **full Procrustes mean***), of a set of preshapes $\{w_i\}$ (of centered configurations $\{Y_i\}$) is the minimizer of the sum of squares of full Procrustes distances from each w_i to an unknown unit size mean configuration μ, i.e.*

$$\begin{aligned} [\hat{\mu}] &= \arg\min_{\mu:||\mu||=1} \sum_{i=1}^{n} \arg\min_{\beta_i,\theta_i,a_i,b_i} D^2(w_i, \mu) = \arg\min_{\mu:||\mu||=1} \sum_{i=1}^{n} D_F^2(w_i, \mu) \\ &= \arg\min_{\mu:||\mu||=1} \sum_{i=1}^{n}(1 - \mu^*w_iw_i^*\mu) = \arg\max_{\mu:||\mu||=1} \mu^*[\sum_{i=1}^{n} *w_iw_i^*]\mu \end{aligned} \tag{5}$$

i.e. $\hat{\mu}$ is given by the complex eigenvector corresponding to the largest eigenvalue of S (Result 3.2 of [3]). Note that all rotations of $\hat{\mu}$ are also solutions but these all correspond to the same shape (equivalence class), $[\hat{\mu}]$.

Shape Variability in Tangent to Shape Space: The structure of shape variability of a dataset of similar shapes can be studied in the tangent hyperplane to shape space at the Procrustes mean of the dataset. The tangent space is a linearized version of shape space at a particular point in shape space which is called the **pole** of tangent projection. We shall consider the tangent projections to the preshape sphere after normalizing for rotation (w.r.t. the pole), which form a suitable tangent coordinate system for shape. Tangent space to shape space is a vector space and Euclidean distance in tangent space is a good approximation to Procrustes distance in the vicinity of the pole. (See chapter 4 of [3] for details).

Definition 7 *The* **Procrustes tangent coordinates** *[3] of a centered configuration, Y, taking μ as the pole, are obtained by projecting $z(Y,\mu)$, the Procrustes fit onto μ of the preshape $w(Y)$, into the tangent space at μ, i.e.*

$$v(Y,\mu) = [I_k - \mu\mu^*]z(Y,\mu) = [I_k - \mu\mu^*]\frac{Y}{s(Y)}e^{j\theta(Y,\mu)}. \tag{6}$$

The inverse of the above mapping (tangent space to centered configuration space) is

$$Y(v,\theta,s,\mu) = [(1-v^*v)^{1/2}\mu + v]se^{-j\theta}. \tag{7}$$

The tangent coordinates lie in a $(2k-4)$-dim subspace of$\Re^{2k}$ or equivalently in a $(k-2)$-dim subspace of $\mathcal{C}^k$.

3 Modeling Shape Dynamics

The changing configuration of a group of moving landmarks (point objects) can be modeled as a moving and deforming shape. The distinction between motion and deformation of a deformable shape is not clear. We separate the motion of a deforming shape into motion of the mean shape and its non-rigid deformations as suggested in [29]. We use the term **"shape activity"** to denote a particular stochastic model for shape deformation learnt from a time sequence of landmark configurations. We define a **"stationary shape activity"** as one in which the mean shape remains constant and the deformation process is stationary. A **"non-stationary shape activity"** is an activity whose mean shape changes with time. We discuss below the shape activity models.

3.1 Stationary Shape Activity: Shape Deformation Model in Tangent Space

A sequence of point configurations from a stationary shape activity (SSA), with small system noise variance, would lie close to each other and to their mean shape (see figure 1(a)). Hence the tangent space at the mean is a good approximate linear space to learn the shape deformation dynamics for a SSA. Given a sequence of configurations with negligible observation noise, $\{Y_{raw,t}\}$, from a SSA we learn its Procrustes mean and evaluate the tangent coordinates of shape (using the Procrustes mean as the pole), as described in the previous section, i.e.

$$\begin{aligned} Y_t &= CY_{raw,t}, \\ s_t &\triangleq s(Y_t) = ||Y_t||, \quad w_t = Y_t/s_t, \\ \mu &= \arg \max_{\mu:||\mu||=1} \mu^*[\sum_t w_t w_t^*]\mu \\ \theta_t &\triangleq \theta(Y_t, \mu) = -\arg(w_t^* \mu), \quad z_t = w_t e^{j\theta_t} \end{aligned} \tag{8}$$

$$v_t \triangleq v(Y_t, \mu) = [I_k - \mu\mu^*]z_t = [I_k - \mu\mu^*]\frac{Y_t e^{j\theta_t}}{s_t} \tag{9}$$

Since the tangent coordinates are evaluated w.r.t. the mean shape of the data, they will have zero mean by construction. We define a linear Gauss Markov model in tangent space to model the shape deformation dynamics, i.e.

$$\begin{aligned} v_t &= A_t v_{t-1} + n_t \\ v_0 &\sim \mathcal{N}(0, \Sigma_{v,0}), \quad n_t \sim \mathcal{N}(0, \Sigma_{n,t}) \end{aligned} \tag{10}$$

Now since the deformation process is assumed stationary, $\Sigma_{v,0} = \Sigma_{v,t} = \Sigma_v$, $\Sigma_{n,t} = \Sigma_n$ and $A_t = A$ is the autoregression matrix and $A < I$ [3]. All three parameters can be learnt using a single training sequence of tangent coordinates, $\{v_t\}$, as follows [30]

$$\begin{aligned} A &= R_v(1)\Sigma_v^{-1} \quad \text{where} \\ \Sigma_v &= \frac{1}{T}\sum_{t=1}^{T} v_t v_t^T \quad \text{and} \quad R_v(1) = \frac{1}{T-1}\sum_{t=2}^{T} v_t v_{t-1}^T \\ \Sigma_n &= \frac{1}{T}\sum_{t=1}^{T}(v_t - Av_{t-1})(v_t - Av_{t-1})^T \end{aligned} \tag{11}$$

and the joint pdf of v_t is given by

$$\begin{aligned} f_t(v_t) &= \mathcal{N}(0, \Sigma_v), \quad \forall t \\ f_t(v_t|v_{t-1}) &= \mathcal{N}(Av_{t-1}, \Sigma_n), \quad \forall t. \end{aligned} \tag{12}$$

Note that the asymptotically stationary case where $A < I$ but $\Sigma_{v,0} \neq \Sigma_v$ so that $\Sigma_{v,t} \neq \Sigma_v$ for the initial few time instants, can also be dealt with in the above framework. In that case $\Sigma_{v,0}$ is defined using a-priori knowledge while $\Sigma_v, R_v(1)$ can be learnt as in (11) but by excluding the summation over the first few time instants.

[3] or equivalently the matrix $(I - A)$ is positive definite

3.2 Partially Observed Shape Dynamics

In the previous subsection we defined a dynamic model on the shape of a configuration of moving points. We assumed that the observation sequence used for learning the shape dynamics is hand-picked so that it has zero (negligible) observation noise associated with it. But a test sequence of point configurations, $\{Y_{raw,t}\}$, will usually be obtained automatically using a measurement algorithm. It will thus have non-negligible observation noise associated with it, i.e. $Y_{raw,t} = Y_{raw,t}^{actual} + \zeta_{raw,t}$ where $\zeta_{raw,t}$ is zero mean Gaussian noise, $\zeta_{raw,t} \sim \mathcal{N}(0, \Sigma_{obs,raw,t})$. Now translation normalization is a linear process and hence $Y_t = CY_{raw,t}$ is also Gaussian with observation noise given by

$$\Sigma_{obs,t} = C\Sigma_{obs,raw,t}C^T \tag{13}$$

(C is the centering matrix defined in (1)). But the mapping from centered configuration space to the tangent to shape space is nonlinear (equation (9)) and hence it is not possible to obtain a closed form expression for the pdf of the tangent coordinates of shape corrupted by observation noise in the configurations. One solution to this problem is to define a partially observed dynamical model with the observed centered configuration (Y_t) forming the observation vector and the shape and motion parameters (scale, rotation) forming the state vector. This is possible to do because there exists an inverse mapping, h, between the state $X_t = [v_t^T, s_t, \theta_t]^T$ and the observation Y_t.

We have the following **observation model** for a "stationary shape activity":

$$\begin{aligned} Y_t &= h(X_t) + \zeta_t, \qquad \zeta_t \sim \mathcal{N}(0, \Sigma_{obs,t}) \\ h(X_t) &= z_t s_t e^{-j\theta_t}, \quad \text{where} \quad z_t = (1 - v_t * v_t)^{1/2}]\mu + v_t \end{aligned} \tag{14}$$

Note that in our current implementation we have assumed Gaussian observation noise, ζ_t, but in general it could be any kind of noise. Defining scale and rotation parameters as the state vector implies that we need to define a prior model for them (motion model). The **motion model** can be defined based on either the motion of the mean shape if it is a moving configuration or based on motion of the measurement sensor if the sensor is moving (for e.g. a moving camera or just an unstable camera undergoing a slight random motion) or a combined effect of both. A camera on an unstable platform, like an unmanned air vehicle (UAV), will have small random x-y motion (translation), motion in z direction (scale change) and rotation about the z axis (rotation angle change). The translation gets removed when centering $Y_{raw,t}$. The scale and rotation can be modeled in this case by using a linear Gauss-Markov model both for log of scale and for the unwrapped rotation angle, i.e.,

$$\begin{aligned}
\log s_t &= \alpha_s \log s_{t-1} + (1-\alpha_s)\mu_s + n_{s,t} \\
\log s_0 &\sim \mathcal{N}(\mu_s, \sigma_s^2), \quad n_{s,t} \sim \mathcal{N}(0, \sigma_r^2) \\
\theta_t &= \alpha_\theta \theta_{t-1} + (1-\alpha_\theta)\mu_\theta + n_{\theta,t} \\
\theta_0 &\sim \mathcal{N}(0, \sigma_\theta^2), \quad n_{\theta,t} \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)
\end{aligned} \tag{15}$$

The motion model parameters can be learnt using the training sequence values of $\{s_t\}_{t=1}^T$ and $\{\theta_t\}_{t=1}^T$ given by (8). $\{\theta_t\}_{t=1}^T$ will have to be the unwrapped value of the rotation angle to learn a Gaussian model. Also, one can either assume wide sense stationarity, in which case $\mu_s, \sigma_s^2, \sigma_r^2, \alpha_s$ and $\mu_\theta, \sigma_\theta^2, \sigma_u^2, \alpha_\theta$ can be learnt using Yule-Walker equations [30], or assume a Brownian motion model(set $\alpha_s = 1$ and $\alpha_\theta = 1$), depending on the dynamics of the application.

The *shape deformation dynamics* (equation (10) in section 3.1) and the *motion model* defined above (equation (15)) form the prior **system model** while equation (14) defines the **observation model**. Thus we have defined a **partially observed dynamic model** for a "stationary shape activity". The model is non-linear since the mapping $h(X_t)$ is nonlinear. Hence we cannot use a Kalman filter to filter out the shape from the noisy observed configurations, and instead we use a particle filter. A particle filter is a sequential Monte Carlo method, that provides at each time instant, t, an n-particle empirical estimate of the posterior probability distribution, $\pi(x_t|y_{0:t}) \stackrel{\triangle}{=} \pi_{t|t}(x_t)$, of the state conditioned on the noisy observations upto the current time t. We describe the particle filtering algorithm in section 3.4 after defining the non-stationary shape activity model in the following subsection.

3.3 Non-stationary Shape Activity

In section 3.1, we considered a stationary shape activity for which the shape remained close to the Procrustes mean, so that one could define a dynamic model for shape deformation in a single tangent space (at the mean). But for a non-stationary shape activity (NSSA), the shape, z_t, is time-varying and so we need to define the dynamics in the time-varying tangent space at the current shape, z_t (see figure 1(b)). The shape dynamics would now include a non-zero shape velocity (time derivative of shape). Denote the shape at time t by z_t and the tangent space at z_t by T_{z_t}. The shape velocity at time t is defined in T_{z_t} which is a vector space approximation to the shape space around z_t.

We define a random walk on the tangent velocity vector coefficients as follows: The tangent hyperplane at any time is a $(k-2)$-dim subspace of $\mathcal{C}^k$ (explained in section 2) and hence any tangent space vector has only $(k-2)$

independent coefficients. Denote tangent velocity by $\Delta v_t \in \mathcal{C}^k$ and its coefficients' vector by $\Delta c_t \in \mathcal{C}^{k-2}$. We define a random walk on Δc_t, i.e. $\Delta c_t = \Delta c_{t-1} + n_t$. The tangent velocity is obtained from Δc_t as $\Delta v_t = U_{t-1}\Delta c_t = \sum_{i=1}^{k-2} \Delta c_{t-1,i} \underline{u}_{t-1,i}$. $U_{t-1} = [\underline{u}_{t-1,1}, \underline{u}_{t-1,2}, ...\underline{u}_{t-1,k-2}]$ where $\{\underline{u}_{t-1,i}\}_{i=1}^{k-2}$ are the eigenvectors corresponding to non-zero eigenvalues of the tangent projection matrix $[I_k - z_{t-1}z_{t-1}^*]C$ and so they form an orthogonal basis for $T_{z_{t-1}}$ [4]. The tangent velocity, Δv_t, is added to the tangent coordinate of z_{t-1} in $T_{z_{t-1}}$ (which is zero by construction) and projected back into the shape space to obtain z_t. Summarizing the dynamics $[z_{t-1}, \Delta c_{t-1}] \rightarrow [z_t, \Delta c_t]$,

$$\begin{aligned} \Delta c_t &= \Delta c_{t-1} + n_t, \quad \text{where} \quad n_t \sim \mathcal{N}(0, \Sigma_n) \\ \Delta v_t &= U_{t-1}\Delta c_t \quad \text{where} \quad U_{t-1} = basis(T_{z_{t-1}}) \\ z_t &= (1 - \Delta v_t^* \Delta v_t)^{1/2} z_{t-1} + \Delta v_t. \end{aligned} \tag{16}$$

As before, for non-negligible observation noise we need a partially observed dynamical model. In this case the **state vector** is $X_t = [\Delta c_t^T, z_t^T, s_t, \Delta\theta_t]^T$. $\Delta\theta_t$ is the angle by which w_t, the preshape at t, needs to be rotated to align it with z_{t-1} [5]. The **observation model** for a NSSA is:

$$Y_t = \tilde{h}(X_t) + \zeta_t, \quad \text{where} \quad \tilde{h}(X_t) = z_t s_t e^{-j\Delta\theta_t}. \tag{17}$$

The **motion model** (model on $s_t, \Delta\theta_t$) can be defined exactly as in equation (15) but replacing θ_t by $\Delta\theta_t$. As before, one can assume the motion model to be stationary or non-stationary. Thus the *shape deformation dynamics* (equations (16)) and the *motion model* (equations (15)) form the prior **system model** in this case and (17) defines the **observation model**.

Training: Given a training sequence of centered configurations, $\{Y_t\}_{t=1}^T$, we obtain a training sequence of $\{\Delta c_t\}_{t=1}^T$ (to learn Σ_n) and of $\{s_t\}_{t=1}^T$, $\{\Delta\theta_t\}_{t=1}^T$ (to learn the motion model parameters) as follows:

$$\begin{aligned} s_t &= ||Y_t||, \quad w_t = Y_t/s_t, \\ \Delta\theta_t &= -\arg(w_t^* z_{t-1}), \quad z_t = w_t e^{j\Delta\theta_t}, \\ \Delta c_t &= {U_{t-1}}^* z_t. \end{aligned} \tag{18}$$

The third equation, $\Delta c_t = {U_{t-1}}^* z_t$, holds because $\Delta c_t = {U_{t-1}}^* \Delta v_t = {U_{t-1}}^*[I - z_{t-1}z_{t-1}^*]z_t = {U_{t-1}}^*[I - z_{t-1}z_{t-1}^*]Cz_t = {U_{t-1}}^* U_{t-1} {U_{t-1}}^* z_t = {U_{t-1}}^* z_t$.

[4] $U_t^{k\times(k-2)} = basis(T_{z_t})$ is evaluated as : $U_t = U_{full,t}Q$ where $U_{full,t}SU_{full,t}^* = [I_k - z_t z_t^*]C$, and $Q = [I_{(k-2)\times(k-2)}, 0_{(k-2)\times 2}]^T$

[5] Rotation normalization w.r.t z_{t-1} before projecting into $T_{z_{t-1}}$

3.4 Particle filtering

Let the initial state distribution be denoted by $\pi_0(dx)$, the state transition kernel by $K_t(x_t, dx_{t+1})$ and the observation likelihood given the state, by $g_t(y_t|x_t)$. The particle filter (PF) [5] is a recursive algorithm which produces at each time t, a cloud of n particles, $\{x_t^{(i)}\}_{i=1}^n$, whose empirical measure closely "follows" $\pi_t(dx_t|y_{0:t})$, the posterior distribution of the state given past observations (denoted by $\pi_{t|t}(dx)$ in the rest of the paper). It starts with sampling n times from the initial state distribution $\pi_0(dx)$ to approximate it by $\pi_0^n(dx) = \frac{1}{n}\sum_{i=1}^n \delta_{x_0^{(i)}}(dx)$ and then **implements the Bayes' recursion** at each time step. Now given that the distribution of X_{t-1} given observations upto time $t-1$ has been approximated as $\pi_{t-1|t-1}^n(dx) = \frac{1}{n}\sum_{i=1}^n \delta_{x_{t-1}^{(i)}}(dx)$, the prediction step samples the new state $\bar{x}_t^{(i)}$ from the distribution $K_{t-1}(x_{t-1}^{(i)}, .)$. The empirical distribution of this new cloud of particles, $\pi_{t|t-1}^n(dx) = \frac{1}{n}\sum_{i=1}^n \delta_{\bar{x}_t^{(i)}}(dx)$ is an approximation to the conditional probability distribution of X_t given observations upto time $t-1$. For each particle, its weight is proportional to the likelihood of the observation given that particle, i.e. $w_t^{(i)} = \frac{n g_t(y_t|\bar{x}_t^{(i)})}{\sum_{i=1}^n g_t(y_t|\bar{x}_t^{(i)})}$. $\bar{\pi}_{t|t}^n(dx) = \frac{1}{n}\sum_{i=1}^n w_t^{(i)} \delta_{\bar{x}_t^{(i)}}(dx)$ is then an estimate of the probability distribution of the state at time t given observations uptil time t. We resample n times with replacement from $\bar{\pi}_{t|t}^n(dx)$ to obtain the empirical estimate $\pi_{t|t}^n(dx) = \frac{1}{n}\sum_{i=1}^n \delta_{x_t^{(i)}}(dx)$. Note that both $\bar{\pi}_{t|t}^n$ and $\pi_{t|t}^n$ approximate $\pi_{t|t}$ but the resampling step is used because it increases the sampling efficiency by eliminating samples with very low weights.

3.4.1 Why Particle Filtering?

We cannot use a Kalman filter in this situation since the system is non-linear. One could use an Extended Kalman Filter (EKF) [6] which linearizes the non-linear system at each time instant using Taylor series and runs a Kalman filter for the linearized system. For the Taylor series approximation to be valid, one requires the initial guess (point about which you linearize) to be close to the actual value at every time instant. Typically linearization is done about the predicted state. This means that one poorly estimated state will cause more error in the next prediction and the error will propagate (thus an EKF cannot come back in track once it loses track). Loss of track can occur due to an outlier observation, modeling error or large system noise. A PF on the other hand can be made asymptotically stable [31, 32] (it can asymptotically track a system even with wrong initial conditions) and hence it gets back in track more easily after losing track (But due to finite number of particles, n, there is only a certain amount of track loss that even a PF can handle).

An EKF (and also a Kalman filter) is unable to track non-Gaussian systems, in particular systems with multimodal priors or posteriors, while a PF can. Multi-modal system models are required to track a sequence of activities (discussed in section 5.2.2) or multiple simultaneous activities. Another point to note is that there are convergence proofs available for convergence of particle filtering estimates of posterior expectations of bounded functions (Chapter 2 of [4]) as the number of particles, n, tends to infinity. Also, the number of particles, n, required to achieve a certain performance gaurantee on estimation error, does not increase with increasing dimension of the state space [4]. The number of particles, n, depends only on the total randomness in the system. So for a system which is more random (larger system noise or observation noise), PF performance can be improved by increasing n, while EKF performance will only deteriorate.

4 Abnormal Activity Detection

An abnormal activity (which may be suspicious behavior) is defined as a change in the system model, which could be slow or drastic, and whose parameters are unknown. Given a test sequence of observations and a "shape activity" model, we define a change detection statistic to detect a change (observations stop following the given shape activity model). We consider only stationary shape activities in this work. The cases of negligible observation noise (Fully Observed) and non-negligible observation noise (Partially observed) are discussed separately.

4.1 Fully Observed Case

The system is said to be fully observed when the observation noise is assumed to be zero (negligible compared to the system noise, n_t). For such a test observation sequence, the shape dynamics of section 3.1 fully defines the "shape activity model". We can evaluate the tangent coordinates of shape (v_t) directly from the observations using (9). We use log-likelihood to test for abnormality. A given test sequence is said to be generated by a *normal activity iff* the probability of occurrence of its tangent coordinates using the pdf defined by (12) is large (greater than a certain threshold). Thus the distance to activity statistic for an '$L+1$' length observation sequence ending at time t, $d_{L+1}(t)$, is the negative log likelihood of the sequence of tangent coordinates of the shape of the observations (first used by us in [33]). We can test for abnormality at any time t by evaluating $d_{L+1}(t)$ for the past $L+1$ frames. $d_{L+1}(t)$ is defined as follows: (K is a constant defined in equation (22))

$$\begin{aligned} d_{L+1}(t) &= -2\log f(v_{t-L}, v_{t-L+1}, ...v_t) \\ &= v_{t-L}^T \Sigma_v^{-1} v_{t-L} + \sum_{\tau=t-L+1}^{t} (v_\tau - Av_{\tau-1})^T \Sigma_n^{-1} (v_\tau - Av_{\tau-1}) + K \end{aligned} \tag{19}$$

4.2 Partially Observed Case

In a partially observed system, the observation noise in the configuration landmarks' measurements is assumed non-negligible and it is defined by the observation model discussed in section 3.2. A PF is used to estimate the posterior distribution of shape at time t given observations upto $t-1$ (prediction) and upto t (filtering). We use these posteriors to define abnormality detection strategies.

1. If the abnormality is a drastic one it will cause the PF, with n large enough to accurately track only normal activities, to lose track. This is because under the normal activity model (equations (10) and (15)), the abnormal activity observations (which do not follow this model) would appear to have a very large observation noise. Thus the tracking error will increase for an abnormal activity (very quickly for a drastic one) and this can be used to detect it. **Tracking error** or prediction error is the distance between the current observation and its prediction based on past observations, i.e.

$$\text{Tracking error} \stackrel{\triangle}{=} ||Y_t - \hat{Y}_t||^2 = ||Y_t - E[Y_t|Y_{0:t-1}]||^2 = ||Y_t - E_{\pi_{t|t-1}}[h(X_t)]||^2 \tag{20}$$

Filtering error, $||Y_t - Y_t^{filt}||^2 = ||Y_t - E_{\pi_{t|t}}[h(X_t)]||^2$ can also be used instead of tracking error

2. For the case when the abnormality is a slow change (say a person walking away slowly in a wrong direction), the PF does not lose track very quickly (tracking error increases slowly) or if it is a short duration change it may not lose track at all. The tracking error will thus take longer to detect the change or it may not detect it at all. For such a case, we propose to use the **expected log likelihood (ELL)** (first proposed by us in [34]) which is the **expectation under the posterior, $\pi_{t|t}$, of the *negative* log likelihood** of the tangent coordinates (in normal system model), i.e.

$$\text{ELL} \stackrel{\triangle}{=} E = E_{\pi_{t|t}}[-log f(v_t)] \tag{21}$$

Note that E is a posterior expectation of the right hand side (RHS) of (19) with $L = 0$. In general, one could use a seqeuence of past shapes ($L > 0$) in this case as well. The expression for E is approximated by E_n as follows

$$E_n \stackrel{\triangle}{=} E_{\pi^n_{t|t}}[-log f(v_t)] = \frac{1}{n}\sum_{i=1}^{n} {v_t^{(i)}}^T \Sigma_v^{-1} v_t^{(i)} + K, \text{ where } K \stackrel{\triangle}{=} \log\sqrt{(2\pi)^{2k-4}|\Sigma_v|}. \tag{22}$$

Now since the PF loses track slowly, the estimated posterior π^n_t remains a good approximation of π_t until the PF has lost track. But a slowly changing shape does introduce a systematically increasing bias in the tangent coordinates of shape (they no longer remain zero mean) and hence the ELL would increase.

To detect any kind of abnormality (slow or drastic) without knowing its rate of change, we use a combination of ELL and tracking error. We declare a sequence of observations to be abnormal when either ELL or tracking error exceeds its corresponding threshold.

5 Tracking, Classification and Activity Sequence Identification

5.1 Tracking Observations

In the previous section, we used a PF in the filtering mode to estimate the probability distribution of shape from noisy observations and used it for abnormality detection. But the PF also provides at each time instant the prediction distribution, $\pi_t(x_t|y_{0:t-1})$, which can be used to predict the expected configuration at the next time instant, i.e. $\hat{Y}_t \triangleq E[Y_t|Y_{0:t-1}] = E_{\pi_{t|t-1}}[h(X_t)]$. We can use this information to improve the measurement algorithm used for obtaining the observations (a motion detector [35] in our case). Its computational complexity can be reduced and its ability to ignore outliers can be improved by using the predicted configuration and searching only locally around it for the currrent observation. As we will show in section 7.1.3, the observed configuration is close to its prediction when there is no abnormality or change. One thing to note here is that in certain cases (for example, if the posterior of any state variable is multimodal), evaluating the average as a prediction of the current observation is not the correct thing to do. In such a case, one tracks observations using the CONDENSATION algorithm [7] which searches for the current observation around each of the possible $h(\bar{x}_t^i), i = 1, 2...n$.

In fact if used in this "tracking observations and filtering" framework, a lot of drastic abnormalities can be detected at the measurement stage itself because no observations will be found in the "vicinity" (region of search defined using observation noise variance) of the predicted position. Also, if the configuration is a moving one, then the predicted motion information can be used to translate, zoom or rotate the camera (or any other sensor) to better capture the scene but in this case, we will have to alter the motion model to include a control input.

5.2 Activity Classification and Sequence Identification

5.2.1 Activity Classification

If the mean shapes of the different shape activities are different, one can simply use the Procrustes distance between the mean shapes to classify them and then track using the correct system model. We consider here the more difficult situation where two or more activities start with the same mean shape but the shape transition models are different.

The particle filtering framework can be adapted easily for classification followed by tracking using the transition model for the chosen activity. We use a discrete model variable [25] as part of the state vector to denote each activity type and make the state transition model a mixture (multimodal) distribution. As discussed earlier, the PF can track multimodal distributions.For classification between P classes, we start with P different modes each having the same mixture probability and within a few frames the PF is able to correctly identify the mode, i.e. posterior probability of the mode given observations converges to the correct mode in a few frames.

5.2.2 Activity Sequence Identification

We consider two possible situations for tracking a sequence of activities. All of these assume that the shape transitions are small enough so that the tangent space is a good approximation to dynamics in shape space. Once again we assume that different activities are identified by differences in their shape transition models.

1. First consider the simple case when there are just two possible activities and their order of occurrence is known, only the change time is unknown. In this case, one could detect the change using the abnormality detection strategy discussed in section 4 and then start tracking it with the second activity's transition model.

2. Now consider the general case when a sequence of activities occur, and we do not know the sequence in which they occur. In this case, we use the same idea as in classification. We track the system with a multimodal system model and keep the mode variable as a state. Whenever a change occurs, it takes the mode variable a few time instants to converge to the correct mode. One could replace the multimodal dynamics with that of the detected mode once the mode variable has converged. In this case we can declare an activity to be abnormal (i.e. neither of the known activity types) if the ELL w.r.t all known models exceeds a threshold.

6 Applications

We have presented in this paper, a generic framework for modeling shape activities and detecting abnormalities which has many potential applications based on what we choose as the "landmark". The "landmark" can be a person and one can learn a shape dynamical model for an activity performed by a group of moving people. In surveillance applications specifically where the people have to follow a defined path, say persons deplaning and walking towards the terminal or persons standing in queues of any shape, this framework can be used to model normal behavior. Any kind of suspicious behavior like a person walking away in an un-allowed direction would get detected as an "abnormal activity". We have experimented with this application and results are presented in section 7.1. The "landmark" can be a moving vehicle and one could model traffic in a certain region as the normal activity and define lane change as the "abnormality". The landmarks could also be the various rigid parts of the human body (head, torso, two halves of each arm and leg, hands, feet). Our framework can be used to learn models for and classify different actions or to track a sequence of actions. Most human actions would constitute "non-stationary shape activities" while people following a path would be "stationary shape activities". Using shape makes the representation invariant to camera zoom, translation or in-plane rotation. In the activities performed by moving people or cars, the number of landmarks (people/vehicles) can be time-varying. We address this issue in the next subsection.

In the applications mentioned above, the observation sensor is typically a video camera. But our framework could be used for point location observations obtained from other kind of sensors as well, for e.g. infrared, acoustic, radar or seismic, and only the observation model would change. Infra-red sensors and radars are more popular for night vision applications and in battlefield scenarios. The "landmark" can also be a robot. Our framework can be extended to control a group of robots to perform a certain task by changing their shape in a certain way. Feedback can be used to control the robots based on their current observed configuration and the desired shape or size. Finally, the framework for 2D shapes can be also extended to 3D shapes.

6.1 Time-Varying Number of Landmarks

All the analysis until now assumes that a configuration of points is represented as an element of $\Re^{2k}$ where k is a fixed number of landmarks. Now we consider what happens when the number of landmarks (here the point objects) is time-varying even though the curve formed by joining their locations remains similar. For example, a group of people (or also a group of vehicles) moving on a certain path with fixed initial and final points but number of people on the path decreases by one when a person leaves and increases by one when someone enters. In such a case, we linearly interpolate the curve by joining the landmark points in a predefined order and then re-sample the interpolated curve to get a fixed number of landmarks. The interpolation problem without any a-priori knowledge or assumptions is, as always, an ill-posed one. We have attempted to use two different schemes which exist in literature- "arc-length re-sampling" and "uniform re-sampling".

In **"arc-length resampling"**, one looks at the curve formed by joining the landmarks in a predefined order, and parametrizes the x and y coordinates by the length, l, of the curve, upto that landmark. Let $[x_t(l), y_t(l)]$ be one-dimensional functions of the curve length and seen this way the discrete landmarks $x_{t,j} = x_t(l_j), y_{t,j} = y_t(l_j), j = 0, 1, ..k_t - 1$ are non-uniformly sampled points from the function $[x_t(l), y_t(l)]$ with $l_0 = 0, l_j^2 = l_{j-1}^2 + (x_{t,j} - x_{t,j-1})^2 + (y_{t,j} - y_{t,j-1})^2$. We linearly interpolate using these discrete points to estimate the function $[\hat{x}_t(l), \hat{y}_t(l)]$ and then re-sample it uniformly at points $\tilde{l}_j = (j-1)L/k, j = 0, 1, ..k-1$ (L is the total length, $L^2 = \sum_j l_j^2$) to get a fixed number, k, of uniformly spaced landmarks. Thus, for every configuration of k_t landmarks, we get a new configuration of uniformly sampled (and hence uniformly spaced) k landmarks, which is an element of $\Re^{2k}$. The linear interpolation and resampling stages can be approximated as a linear transformation, B_t (a $k_t \times k$ matrix), applied to the original points. The covariance of observation noise in the re-sampled points becomes $\Sigma_{obs,t}^k = B_t \Sigma_{obs,t}^{k_t} B_t^T = B_t C^{k_t} \Sigma_{obs,raw,t}^{k_t} {C^{k_t}}^T B_t^T$.

"Uniform resampling", on the other hand, assumes that the observed points are uniformly sampled from some process, $[x_t(s), y_t(s)]$, i.e. it assumes that the observed points $x_{t,j} = x_t(s_j), y_{t,j} = y_t(s_j)$ with $s_j = (j-1)/k_t$. We linearly interpolate to estimate $[\hat{x}_t(s), \hat{y}_t(s)]$ and re-sample it uniformly at points $\tilde{s}_j = (j-1)/k$, to get a fixed number of landmarks, k. Assuming the observed points to be uniformly sampled makes this scheme very sensitive to the changing number of landmarks. Whenever the number of landmarks changes, there is a large change in the re-sampled points' configuration. This leads to more false alarms while performing abnormal activity detection. But since this scheme gives equal importance to all observed points irrespective of the distance between consecutive points, it is more quick to detect abnormalities in shape caused even by two closely spaced points. We discuss an example situation in section 7.1.7.

7 Experimental & Simulation Results

7.1 Abnormality Detection

7.1.1 Dataset

We have used a video sequence of passengers deplaning and walking towards the airport terminal as an example of a "stationary shape activity". The number of people in the scene varies with time. We have resampled the curve formed by joining their locations using "arc-length resampling" as described in section 6.1. We needed observation noise-free data to learn the system model and hence we used hand-marked passenger locations for training. The mean shape, μ, and the tangent space Gauss Markov model parameters, A, Σ_v, Σ_n, were learnt using this data (as discussed in section 3.1). Also the motion model parameters (which in this case model random motion of the camera) were estimated with this data.

7.1.2 Experiments

The data, provided to us by the Transport Security Administration (TSA), did not have any instances of abnormal behavior. Abnormal behavior was simulated by making one of the persons walk away in an abnormal direction (in the results shown one person was made to walk away at an angle of 45^o to the X-axis, see figure 2(b); 2(a) shows a normal activity frame). Now, the person could be moving away at any speed which will make the abnormality a slow or a drastic change. We have simulated this by testing for walk away speeds of $1, 2, 4, 16, 32$ pixels per time step in both x and y directions. The average speed of any person in the normal sequence is about 1 pixel per time step. Thus walk-away velocity of 1 pixel per time step, denoted as $vel. = 1$, corresponds to a slow change which does not go out of track for a long time while $vel. = 32$ is a drastic change that causes the PF to lose track immediately. We first show some results with simulated observation noise (i.i.d. Gaussian noise added to the hand-marked points) and then with real observations of the passengers' locations in each frame obtained using a motion detection algorithm described in [35]. The ability of our algorithm to deal with temporal abnormalities [2] is demonstrated in section 7.1.7.

7.1.3 Tracks

Figure 3 (a), (b) show two normal observation frames (circles) and the corresponding tracked configuraton (stars), for simulated observation noise. Figure 4(a),(b) are the two normal observation frames for real observations. This shows the ability of our model to be used for "tracking observations". Since we are using only a point object abstraction for moving objects (here persons), we show observed and tracked point object locations only without showing the actual images.

7.1.4 ELL versus Tracking Error: Slow and Drastic Changes

Figure 3(c) shows a slow abnormality ($vel. = 1$) introduced at $t = 5$ which is tracked correctly for a long time (figure 5(b) plots the tracking error) and hence we need to use ELL to detect it (ELL plots shown in figure 5(a)). Figure 3(d) shows a drastic abnormality ($vel. = 32$) which was also introduced at $t = 5$ but loses track immediately. In this case the abnormal observations are ignored and the PF continues to follow the system model.

As a result, the ELL (figure 5(a)) confuses it for a normal sequence while the tracking error (figure 5(b)) detects it immediately. All of the above plots were obtained by adding observation noise with variance $\sigma^2 = 9$ (3-pixel observation noise) to x and y coordinates of each point. As can be seen from figure 5(a), with $vel. = 1$, the abnormality (introduced at $t = 5$) gets detected at $t = 13$ and with $vel. = 4$ it gets detected at $t = 7$, using ELL and a detection threshold of 4. For $vel. = 32$, the abnormality gets detected immediately using tracking error (figure 5(b)) while ELL is unable to detect it. This demonstrates the need to use a combination of ELL and tracking error to detect both slow and drastic changes (since the aim is to be able to detect any kind of abnormality and its rate of change is not known). As explained earlier, we declare an abnormality if either of ELL or tracking error exceeds their corresponding thresholds.

7.1.5 ROC curves and Performance Degradation with increasing Observation Noise

The above intuition is captured numerically in the ROC (Receiver Operating Characteristic) curves [30, 27] for change detection using ELL (figure 7(a) and (b) for slow and drastic changes respectively), using tracking error (figure 8(a) and (b)) and using a combination of both (figure 9(a),(b),(c),(d)). Please note that every figure of the ROC plot has a different y axis range. The blue circles, red stars, magenta triangles and cyan diamonds are the ROC plots for observation noise variances $= 3, 9, 27, 81$. ROC for a change detection problem [27] plots the average detection delay against the mean time between false alarms by varying the detection threshold. The aim of an ROC plot is to choose an **operating point** threshold which minimizes detection delay and maximizes the mean time between false alarms. Hence we are most interested in minimum detection delay at the rightmost point of the curve.

For the slow change ($vel. = 1$), detection delay is much lesser using ELL than using tracking error while the opposite is true for the drastic change ($vel. = 32$). Detection performance degradation of ELL for slow change and of tracking error for drastic change with increasing observation noise is slow. In figure 7(a), detection delay is 2 time units for $\sigma^2_{obs} = 3$ and 7 time units for $\sigma^2_{obs} = 81$. In figure 8(b), detection delay is 3 time units for $\sigma^2_{obs} = 3$ and 4 time units for $\sigma^2_{obs} = 81$. In figure 9, we plot the ROC curves for change detection using a combination of ELL and tracking error. In this case, for each observation noise variance, there are multiple curves, since one needs to vary thresholds for both ELL and tracking error to get the ROC. A single curve is for the ELL threshold fixed and tracking error threshold varying. We have a set of curves for varying ELL thresholds. We plot low and high observation noise in two separate plots. As can be seen, the combined strategy has better performance than either ELL or tracking error for all rates of change and for all observation noises.

7.1.6 Real Observations

The trends discussed above are also observed for real observations obtained using a motion detector [35] on the image sequences. The trends are not as clear though since observation noise model assumptions are not followed exactly. Observation noise was modeled to be Gaussian (although the PF can filter non-Gaussian noise as well) and its covariance was learnt from a training sequence of observations obtained using the motion detector. The tracks, ELL plots and tracking error plots are shown in figures 4, 6(a) and 6(b) respectively. Abnormality was again introduced at $t = 5$. The description of the plots is same as that for simulated noise discussed above.

7.1.7 Temporal abnormality

We also tested our framework for detecting what is referred to in [2] as a temporal abnormality (one person stopped in his normal path). It gets detected in this framework because there is a change in shape when the person behind the stopped person goes ahead of him (curve becomes concave). We used “uniform resampling” (discussed in section 6.1) which detected temporal abnormality easily using ELL (figure 10). “Arc-length resampling” does not work too well in this case. This is because it tends to average out the locations of two closely spaced points, thus smoothing out the concavity which needs to be detected. “Uniform resampling”, on the other hand, assumes the observed points are uniformly sampled and hence gives equal weight to all observed points irrespective of the distance between them. Thus it is able to detect concavity caused even by two closely spaced points. Another way to detect temporal abnormality would be to use a non-stationary shape activity model and look at deviations from the expected value of shape velocity.

7.2 Activity Classification and Sequence Identification

7.2.1 Simulated Shape

To test the applicability of our approach to shape activity classification and to activity sequence identification, we used simulation data. We used a regular pentagon as the pole for tangent space projections and specified two different system models in tangent space to represent two shape activities. We consider activites which have different transition models in tangent space (but the changes are assumed small enough so that a single tangent space remains a valid approximation to shape space). The two activities we considered had the following transition models:

$$\text{Activity 1:} \quad v_t = Av_{t-1} + n_t + 0, \qquad \text{Activity 2:} \quad v_t = Av_{t-1} + n_t + b \tag{23}$$

7.2.2 Activity Classification

Figure 11 (a) and (b) show the posterior model probabilities as time progress for Activity 1 and Activity 2 respectively. In both cases, the model probability converges to its correct value by $t = 12$.

7.2.3 Activity Sequence Identification

We simulated the two situations discussed in section 5.2.2. The results are presented below

1. We used the activity models defined in section 7.2.1, the sequence started Activity 1 and transitioned to Activity 2 at $t = 5$. We assumed that only the change time is unknown in this case but the sequence (Activity 1 followed by 2) is known. ELL was able to detect the change to Activity 2 at $t = 11$.

2. We again used the two activity models from section 7.2.1, but with multiple transitions between models and the sequence of transitions not known. The sequence started in Activity 1, then at $t = 5$ it transitioned to Activity 2 and at $t = 16$ it transitioned back to Activity 1. The time between $t = 5$ and $t = 10$ and then between $t = 16$ to $t = 20$ is roughly a transition phase between the two models. Figure 12(a) shows the posterior expected model value (which is directly proportional to the posterior model probability, for the case of two models) and its rounded off value (equivalent to choosing between the models based on higher posterior model probability). As can be seen, the prior expected model value detects the model changes correctly but with some delay. The ELL of the posterior w.r.t. the prior model probability (plotted in figure 12(b)) is also able to identify the models but with some delay.

8 Conclusions and Future Directions

In this paper we have presented a "shape activity model", which is a partially observed dynamical model for the changing configuration of a set of moving landmarks. The shape and motion dynamics constitute the system model and the observed landmark locations form the observation vector. The mapping between state and observation space is nonlinear and hence the shape is estimated from the noisy observations using a particle filter. Abnormal activity detection is formulated as a change detection problem with change parameters being unknown. We have proposed a change detection strategy using particle filters which can detect both slow and drastic changes in partially observed nonlinear systems. Experimental results have been shown for abnormal activity detection in an airport scenario.

The particle filter can also be used for "tracking observations". This has been discussed in section 5.1 and we hope to implement joint tracking and abnormality detection as part of future work. We also hope to classify between different activities and track a sequence of real activities (as discussed in section 5.2). Also, in this work, we have experimented only with stationary shape activities. We are currently studying the non-stationary case (discussed in section 3.3) in more detail. We hope to characterize (define a pdf for) specific instances of a normal activity in the non-stationary case and to define the abnormality detection problem. The non-stationary shape activity model provides the flexibility to model and track a much larger class of group activites. We are also working on providing a theoretical justification for our proposed change detection strategy and proving convergence of the ELL estimate as the number of particles in PF tends to infinity.

The issue of time-varying number of landmarks needs to be studied more rigorously by first defining the optimality criterion to make the interpolation problem well-posed and then deciding the optimal strategy. Also, the current shape space ($\Re^{km}$ modulo Euclidean similarity transformations) can be replaced by general shape spaces. For example, the affine shape space (chapter 12 of [3]) would be useful to make the activity invariant to an affine camera's motion. Finally, our framework can be applied to the different applications (discussed in section 6).

References

[1] D.G. Kendall, D. Barden, T.K. Carne, and H. Le, *Shape and Shape Theory*, John Wiley and Sons, 1999.

[2] W.E.L. Grimson, L. Lee, R. Romano, and C. Stauffer, "Using adaptive tracking to classify and monitor activities in a site," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 22–31.

[3] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis*, John Wiley and Sons, 1998.

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

[5] 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)*, pp. 140(2):107–113, 1993.

[6] T. Kailath, A.H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2000.

[7] J.P. MacCormick and A. Blake, "A probabilistic contour discriminant for object localisation," *IEEE International Conference on Computer Vision*, 1998.

[8] C.T. Zahn and R.Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Transactions on Computers*, vol. C-21, pp. 269–281, 1972.

[9] David F. Rogers and J. Alan Adams, *Mathematical Elements for Computer Graphics*, WCB/McGraw-Hill, 1990.

[10] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active shape models: Their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, January 1995.

[11] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Training Models of Shape from Sets of Examples," in *British Machine Vision Conference*, 1992, pp. 9–18.

[12] J.T. Kent, "The complex bingham distribution and shape analysis," in *Journal of the Royal Statistical Society, Series B*, 1994, pp. 56:285–299.

[13] Y. Zhou, L. Gu, and H. Zhang, "Bayesian tangent space model: Estimating shape and pose parameters via bayesian inference," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[14] A. Srivastava, "Prior models for bayesian filtering in subspace tracking," in *First IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2000.

[15] A. Chiuso and S. Soatto, "Monte-carlo filtering on lie groups," in *IEEE Conference on Decision and Control*, 2000.

[16] S. Kurakake and R. Nevatia, "Description and tracking of moving articulated objects," in *International Conference on Pattern Recognition*, 1992, pp. I:491–495.

[17] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber, "Automatic symbolic traffic scene analysis using belief networks," in *National Conference on Artificial Intelligence*, 1994, pp. 966–972.

[18] T. Starner and A. Pentland, "Visual recognition of american sign language using hidden markov models," in *Proc. Intl. Workshop on Face and Gesture Recognition*, 1995.

[19] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 568–574.

[20] A. Bobick and Y. Ivanov, "Action recognition using probabilistic parsing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[21] A. Roy Chowdhury and R. Chellappa, "A factorization approach for event recognition," in *CVPR Event Mining Workshop*, 2003.

[22] L. Torresani and C. Bregler, "Space-time tracking," in *European Conference on Computer Vision*, 2002.

[23] L. Zelnik-Manor and M. Irani, "Event based analysis of video," in *IEEE International Conference on Computer Vision*, 2001.

[24] D. Ponceleon T. Syeda-Mahmood, "Recognizing action events from multiple viewpoints," in *IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

[25] Shaohua Zhou and Rama Chellappa, "Probabilistic human recognition from video," in *European Conference on Computer Vision*, 2002, pp. 681–697.

[26] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2001.

[27] M. Basseville and I Nikiforov, *Detection of Abrupt Changes: Theory and Application*, Prentice Hall, 1993.

[28] B. Azimi-Sadjadi and P.S. Krishnaprasad, "Change detection for nonlinear systems: A particle filtering approach," in *American Control Conference*, 2002.

[29] S. Soatto and A.J. Yezzi, "Deformotion: Deforming motion, shape average and the joint registration and segmentation of images," in *European Conference on Computer Vision*, 2002, p. III: 32 ff.

[30] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, Inc., 1991.

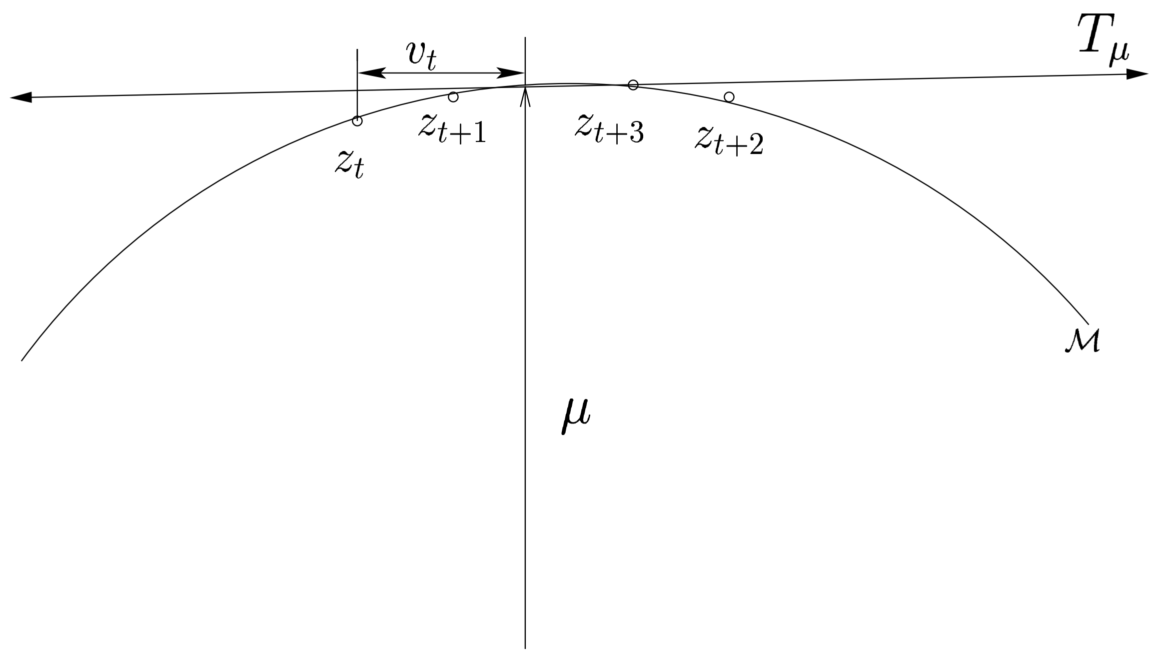
[31] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering for practitioners," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002.

[32] LeGland F. and Oudjane N., "Stability and Uniform Approximation of Nonlinear Filters using the Hilbert Metric, and Application to Particle Filters," *Technical report, RR-4215, INRIA*, 2002.

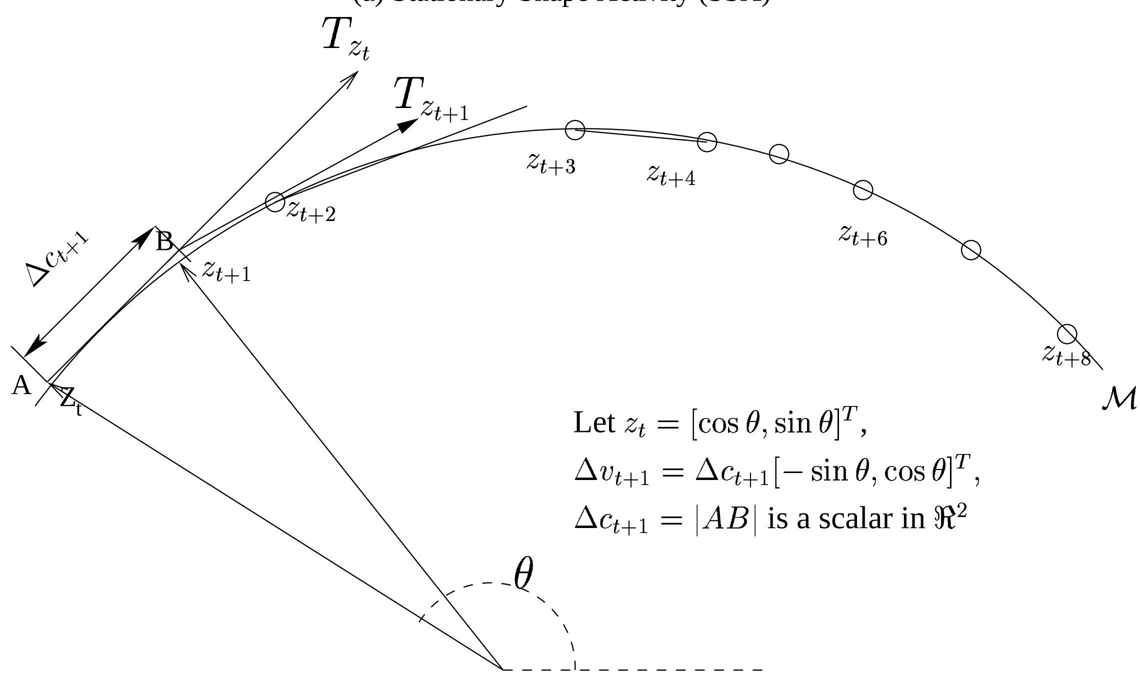
[33] N. Vaswani, A. RoyChowdhury, and R. Chellappa, "Statistical shape theory for activity modeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003.

[34] N. Vaswani, A. RoyChowdhury, and R. Chellappa, "Activity recognition using the dynamics of the configuration of interacting objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[35] Q Zheng and S. Der, "Moving target indication in lras3 sequences," in *5th Annual Fedlab Symposium College Park MD*, 2001.

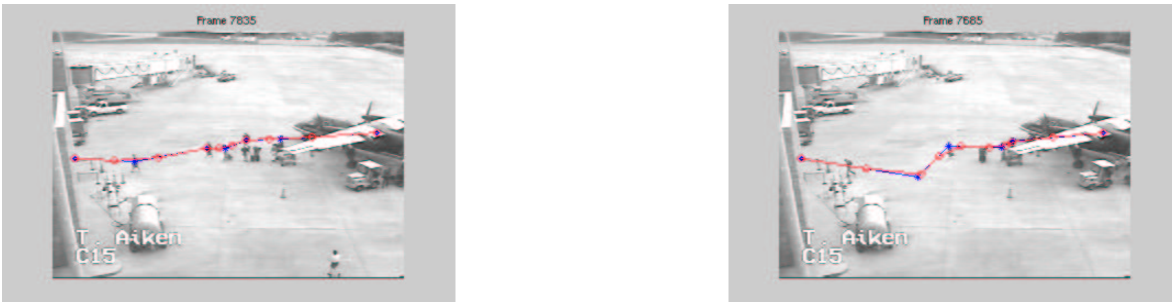


(a) Stationary Shape Activity (SSA)



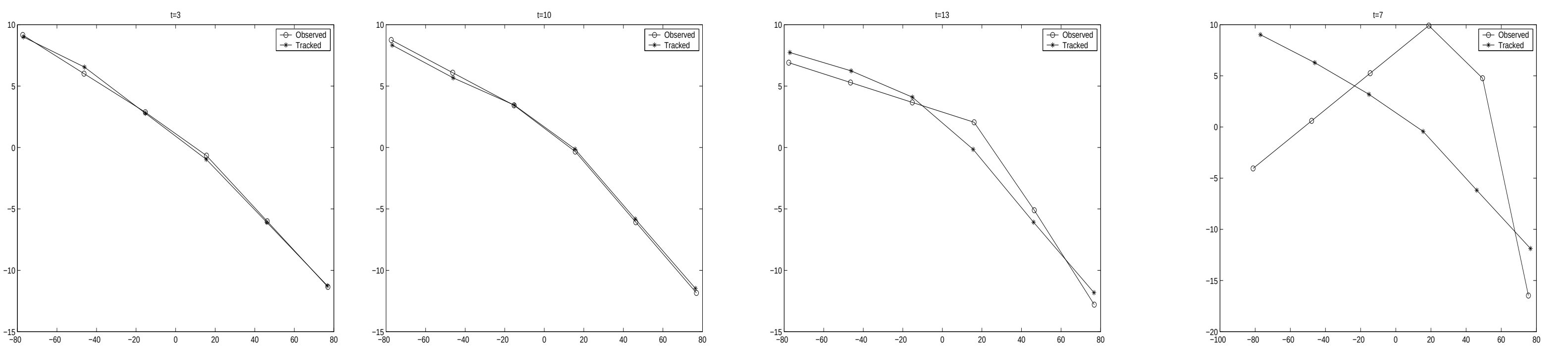
(b) Nonstationary Shape Activity (NSSA)

Figure 1: Stationary and Non-stationary Shape Activities depicted in $\Re^2$. $\mathcal{M}$ denotes the shape space. In (a), we show a sequence of shapes from a SSA; at all times the shapes are close to the mean shape and hence the dynamics can be approximated in T_μ (tangent space at μ). In (b), we show a sequence of shapes from an NSSA, the shapes move on the shape manifold, $\mathcal{M}$, and hence we need to define a new tangent space at every time instant. Δv_{t+1} is the vector AB and Δc_{t+1} is its magnitude.



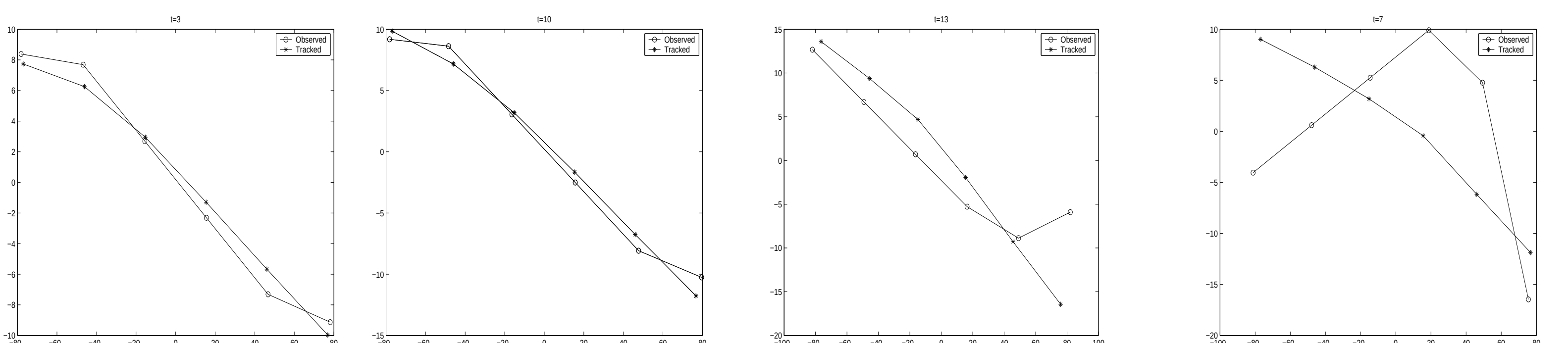
(a) A ‘normal activity’ frame with 4 people (b) Abnormality introduced by making one person walk-away in an abnormal direction

Figure 2: Airport example : Passengers deplaning



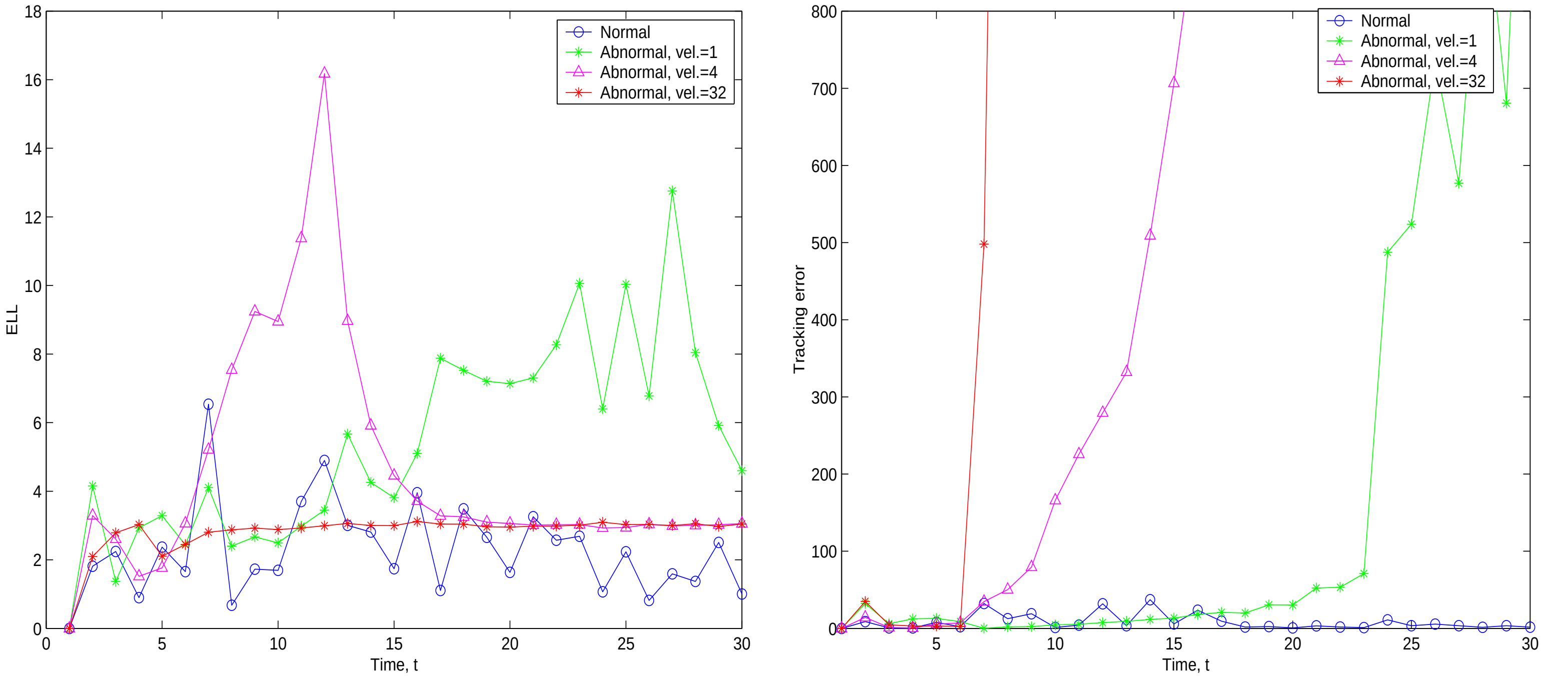
(a)Normal frame: In track (b)Normal frame: In track (c)Slow abnormality: Still in track (d)Drastic abnormality: Loses track

Figure 3: Tracks: Simulated Observation noise, $\sigma^2_{obs} = 9$ (3-pixel noise)



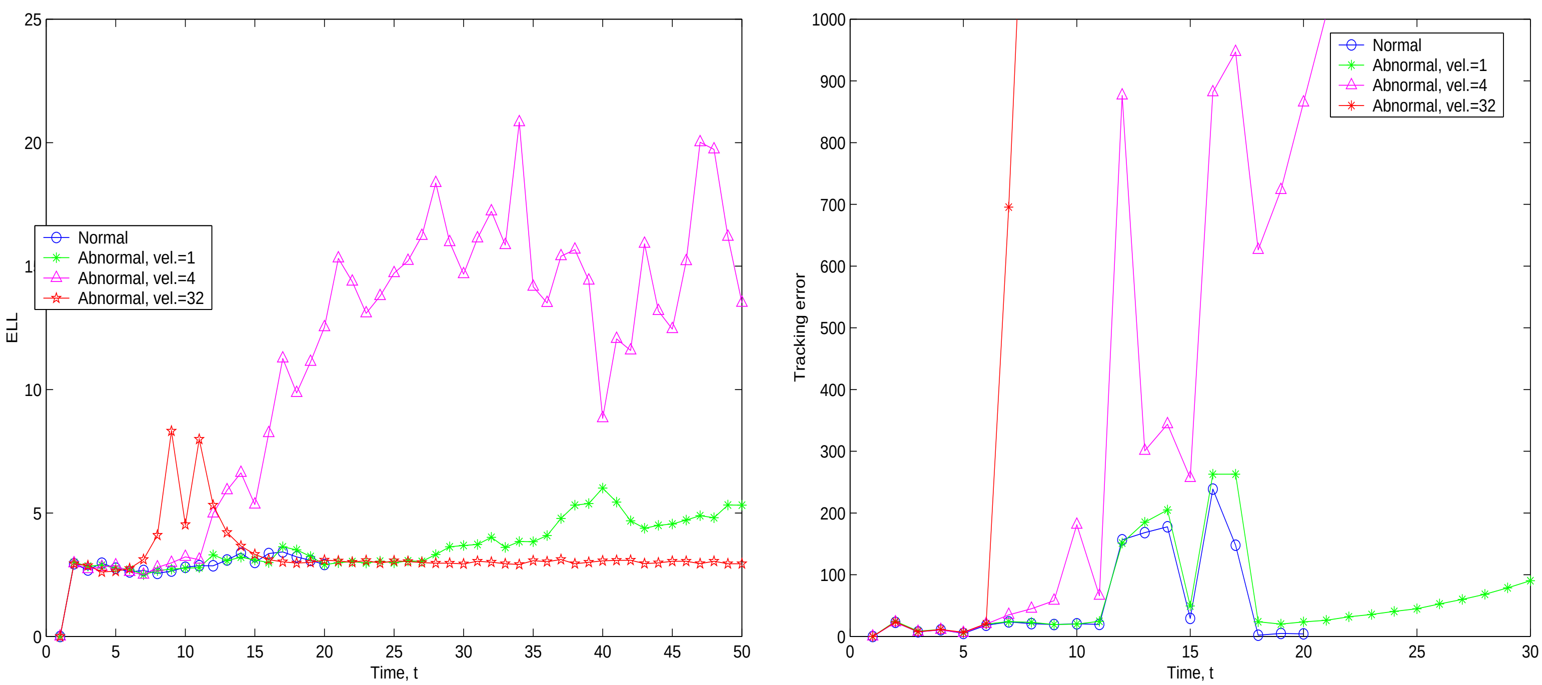
(a)Normal frame: In track (b)Normal frame: In track (c)Slow abnormality: Still in track (d)Drastic abnormality: Loses track

Figure 4: Tracks: Real Observations



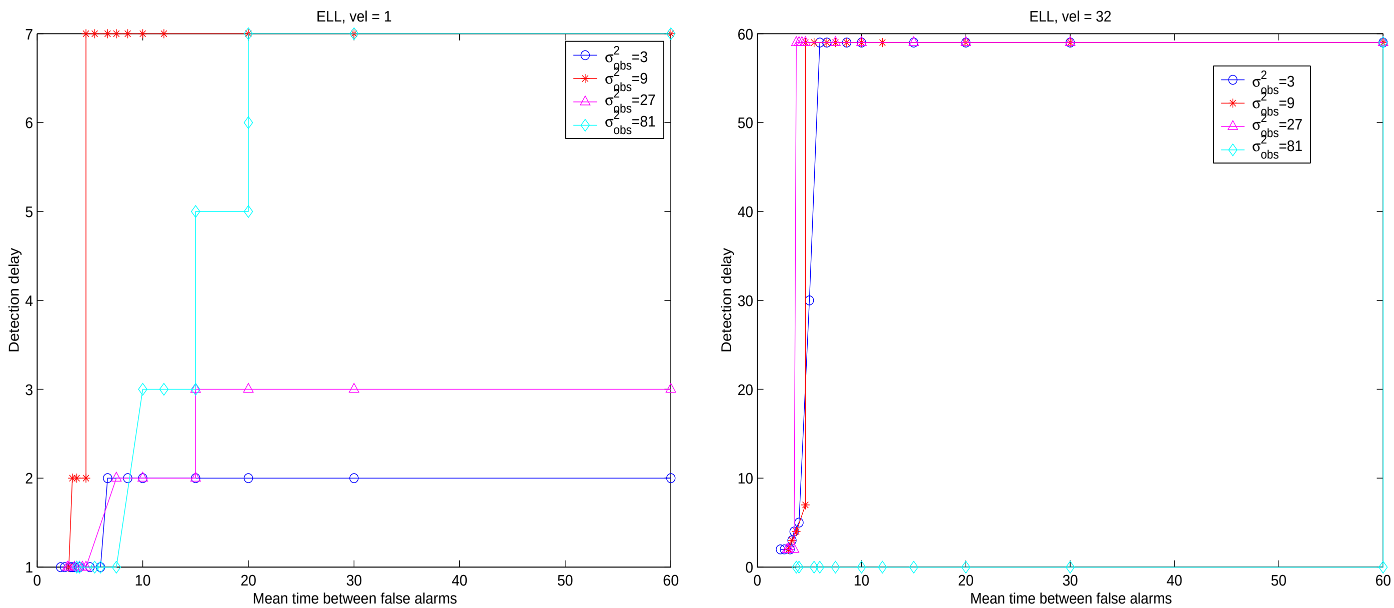
(a) ELL (b) Tracking error

Figure 5: ELL and Tracking error plots: Simulated Observation noise, $\sigma^2_{obs} = 9$ (3-pixel noise). Abnormality was introduced at $t = 5$. As explained in the text, ELL is able to detect slow changes better while tracking error works better for drastic changes.



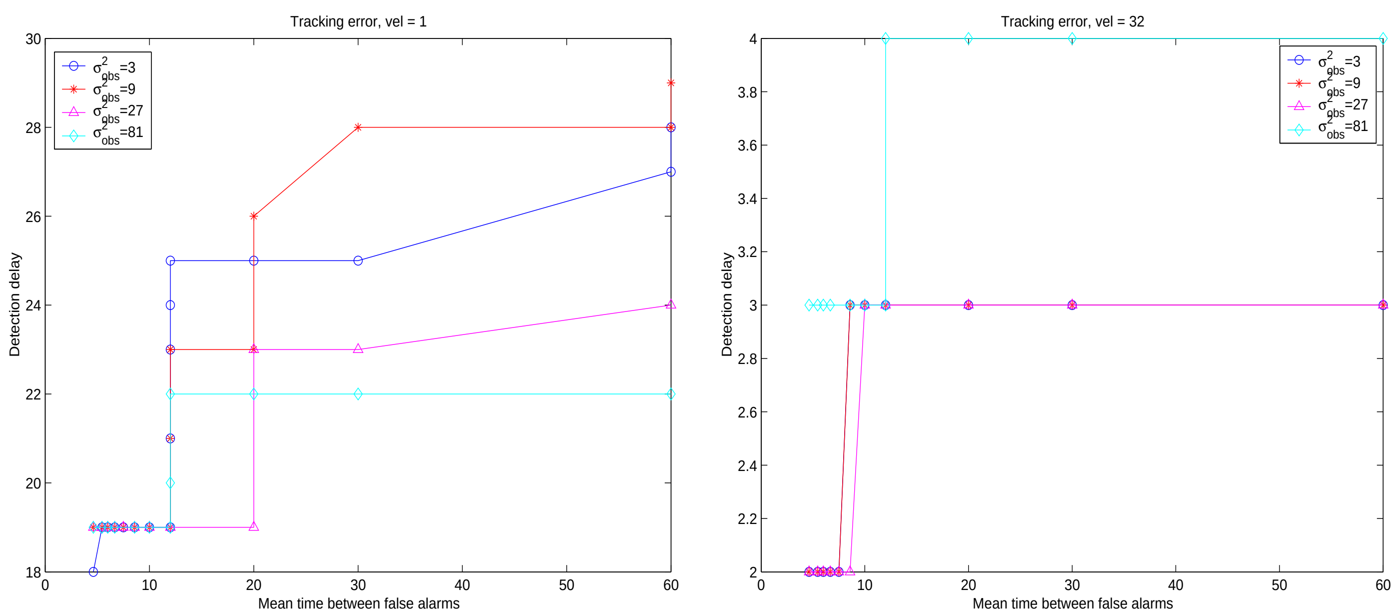
(a) ELL (b) Tracking error

Figure 6: ELL and Tracking error plots: Real Observations. Abnormality was introduced at $t = 5$.



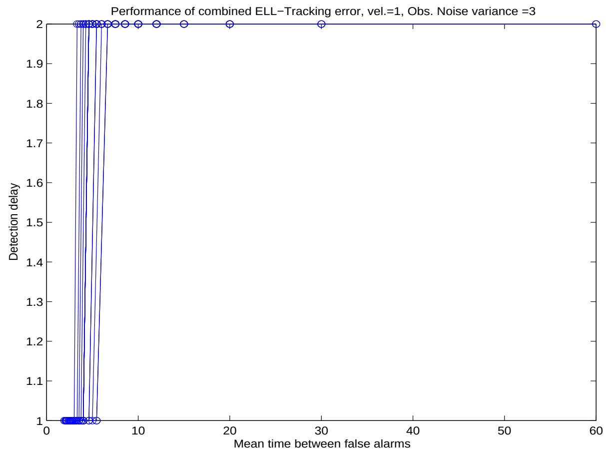
(a) Slow change (vel.=1): **WORKS** (b) Drastic change (vel.=32): DOES NOT WORK

Figure 7: ROCs for Change detection using ELL. Blue circles, red stars, majenta triangles and cyan diamonds plots are for $\sigma_{obs}^2 = 3, 9, 27, 81$ respectively. Please note that the two plots have different y axis ranges. ELL does not work for drastic changes (detection delays in (b) are very large).

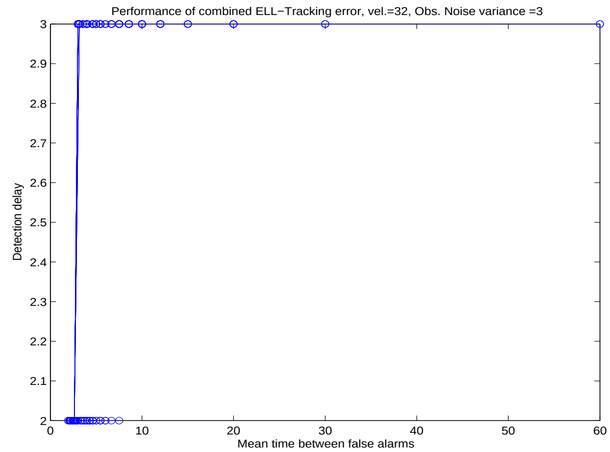


(a) Slow change (vel.=1): DOES NOT WORK (b) Drastic change (vel.=32): **WORKS**

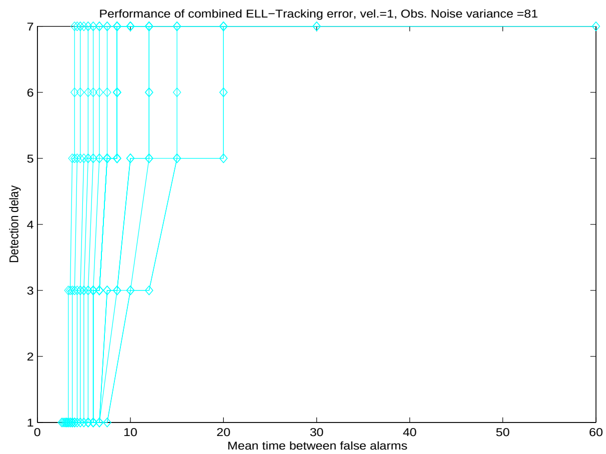
Figure 8: ROCs for Change detection using Tracking error. Blue circles, red stars, majenta triangles and cyan diamonds plots are for $\sigma_{obs}^2 = 3, 9, 27, 81$ respectively. Please note that the two plots have different y axis ranges. Tracking error does not detect slow changes easily (Detection delays in (a) are large).



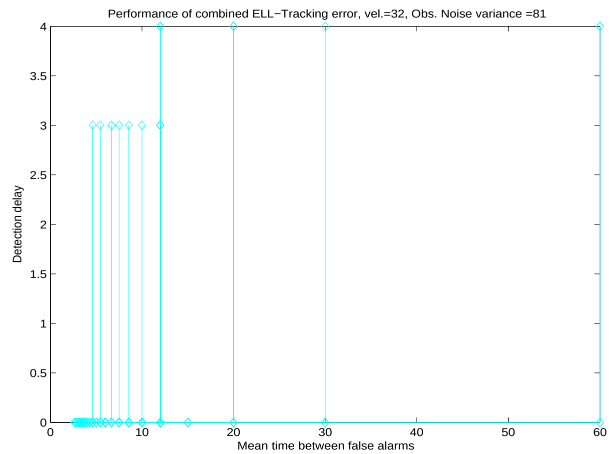
(a) $\sigma^2_{obs} = 3$, Slow change (vel.=1): **WORKS**



(b) $\sigma^2_{obs} = 3$, Drastic change (vel.=32): **WORKS**



(c) $\sigma^2_{obs} = 81$, Slow change (vel.=1): **WORKS**



(d) $\sigma^2_{obs} = 81$, Drastic change (vel.=32): **WORKS**

Figure 9: ROCs for Change detection using Combined ELL-Tracking error. In this case, for each observation noise variance, there are multiple curves, since one needs to vary thresholds for both ELL and tracking error to get the ROC. A single curve is for the ELL threshold fixed and tracking error threshold varying. We have a set of curves for varying ELL thresholds. As discussed in the text, the operating point is the rightmost corner point. The operating point detection delay is 2 and 3 time units for $\sigma^2_{obs} = 3$ ((a) and (b)), and 7 and 4 time units for $\sigma^2_{obs} = 81$ ((c) and (d)).

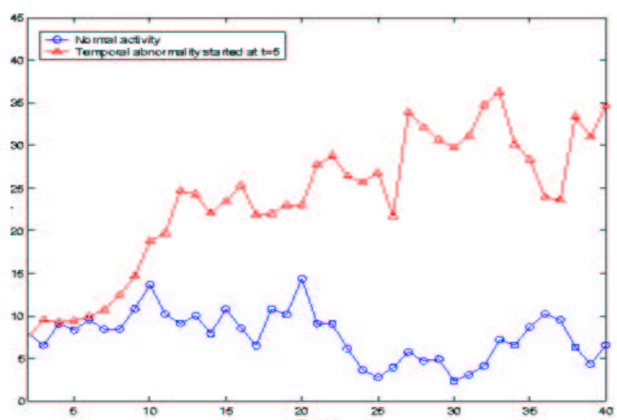
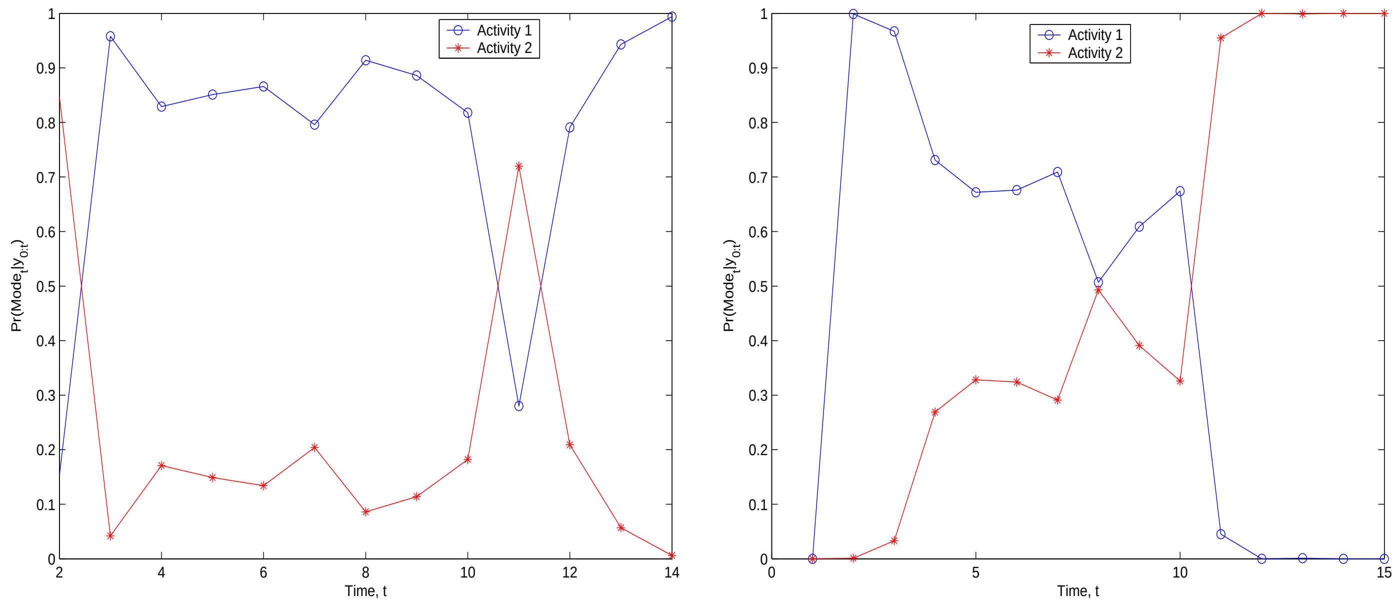
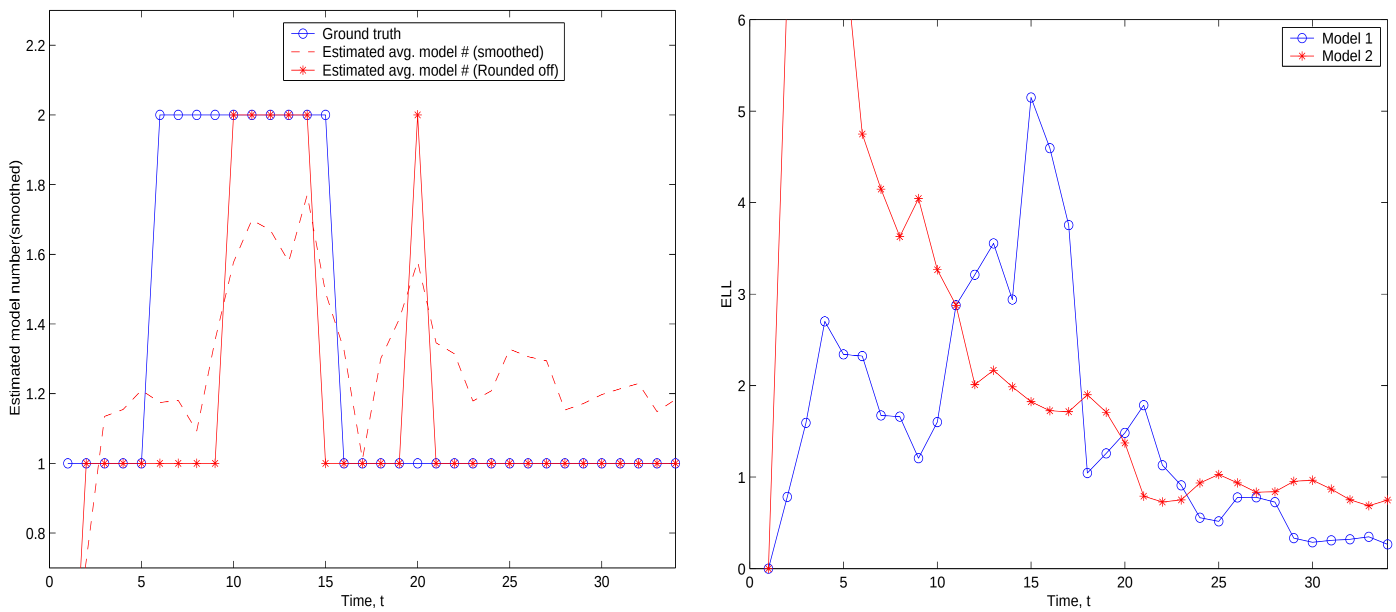


Figure 10: ELL plot for Temporal abnormality detection. Abnormality was introduced at $t = 5$.



(a) Plots of $Pr(Model = i|y_{0:t}^{Activity1}), i = 1, 2$ (b) Plots of $Pr(Model = i|y_{0:t}^{Activity2}), i = 1, 2$

Figure 11: Activity Classification: (a) is for observations belonging to Activity 1, (b) is for observations belonging to Activity 2.



(a) Posterior Expected model number (b) ELL w.r.t model 1 (blue) and model 2 (red)

Figure 12: Tracking a sequence when its order of occurrence is not known. The sequence started in activity 1, changed to activity 2 at $t = 5$ and transitioned back to activity 1 at $t = 16$. Expected model number detects the first change at $t = 10$ and second at $t = 16$ with two errors at $t = 15, 20$. ELL (expected negative log-likelihood) detects the first change (from Activity 1 to Activity 2) at $t = 11$ and the second one (from Activity 2 to Activity 1) at $t = 24$. Betweem $t = 17$ to $t = 24$ (which is roughly the transition period from Activity 2 to 1), ELL confuses between the two models.