A Variant of Particle Filtering using Historic Datasets for Tracking Complex Geospatial Phenomena

Anand Panangadan
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr, Pasadena, CA 91101
1-818-354-3054

Anand.V.Panangadan@jpl.nasa.gov

Ashit Talukder
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr, Pasadena, CA 91101
1-818-354-3054

Ashit.Talukder@jpl.nasa.gov

ABSTRACT

The paper presents an extension of the particle filtering algorithm that is applicable when an accurate state prediction model cannot be specified but a database of prior state evolution tracks is available. The conventional particle filtering algorithm represents the belief state as a collection of particles, where each particle is a sample from the state space. The particles are updated by applying the state space equations. In the proposed approach, each particle is an instance of a complete state trajectory, drawn from the database of historic state trajectories. An explicit state update model is not required as the trajectory represented by each particle is covers the entire modeling time period. When new observations become available, a proportion of the particles are replaced using trajectories from the database, selected based on distance from the observation. This tracking algorithm is applicable where the state evolves in a complex manner as in the eye of tropical cyclones. The proposed technique is evaluated by tracking selected cyclones from 2005 using a database of cyclone tracks from the previous 25 years.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: *Probabilistic algorithms*; I.6.8 **[Types of Simulation]:** *Monte Carlo*

General Terms

Algorithms, Design.

Keywords

Particle filter, Tropical cyclone, State estimation.

1. INTRODUCTION

Particle filtering has been extensively used for tracking in a variety of applications. The typical implementation of this technique requires the specification of a stochastic state model (quantifying the expected evolution of the state over time) and an observation model (quantifying the probability of receiving a specific observation from a given state). In applications involving the tracking of complex natural phenomena, it may not be possible to derive an accurate state model. However, historical records of the time evolution of the phenomenon may be available.

For instance, the path of a cyclone is affected by several factors such as latitude, sea temperature, and presence of land. Thus it is difficult to formulate a concise representation of the state equation for the problem of cyclone tracking. However, the complete cyclone trajectories for the past several decades are available (for instance, from the National Oceanic and Atmospheric Administration, NOAA). The problem of predicting human movement is also difficult owing to the inherent complexity of human behavior. Thus, applications that need to predict human movement are based on learning a model from a set of prior movement tracks.

We present a variation of the particle filtering algorithm that uses such historical records instead of an explicit stochastic state model. The basic idea is that each "particle" represents an entire state sequence from the database of historic state trajectories. An index into the state trajectory indicates the current estimate within the complete track. A state update equation is not required since the particle can be updated at every time-step by just incrementing the index. Instead of updating weights with every new observation by computing the posterior probability, we use a distance metric between each new observation and every particle to resample a proportion of the particles, i.e., some particles are replaced with complete tracks from the database that are closer to the newest observation. The mean of the state trajectories represented by all the particles gives the state estimate at every time-step. As this state estimate is a mean of complete state trajectories, the prediction horizon can be very long. This is the major advantage of our approach. We call our algorithm a data-centric approach to distinguish it from the conventional particle filtering algorithm in this paper.

We apply this technique to the problem of hurricane tracking from satellite imagery. Accurate predictive models of the expected path of a hurricane are not yet available as hurricane evolution is a complex phenomena affected by a number of environmental factors. We show how the historical records of hurricane paths can be used instead of a predictive state model within the particle filtering framework. In our experiments, we utilized tracks recorded in the hurricane seasons of the North Atlantic basin between years 1980 and 2004 (the database) for the predictive tracking of hurricanes in another season (year 2005). We compare the results of hurricane tracking using the proposed data-centric

ACM SIGSPATIAL GIS'10, November 2–5, 2010, San Jose, California, USA.

filtering algorithm with tracks predicted from a particle filter formulated with a first-order velocity model of hurricane movement with random errors.

Another approach to this problem is to explicitly learn a model from the historical record and then use the learned model in a conventional filter. This approach requires that a specific learning mechanism suitable to the application domain be determined. In addition, the model would have to be updated every time a new record is entered into the database unlike in our proposed algorithm.

The format of the paper is as follows. Section 2 describes related work. Section 3 reviews the conventional particle filtering algorithm. The proposed data-centric approach is described in Section 4. Section 5 describes the cyclone tracks that will be used to evaluate the algorithm. Section 6 gives results from our experiments to evaluate the method. Conclusions and directions for future work are in Section 7.

2. RELATED WORK

Modeling complex spatial tracks has been most studied by the vision and robotics research communities for people tracking. Mikami et al. [1] formulated a memory-based particle filter for tracking complex phenomena. Similar to our formulation, the state distribution is specified from long-term dynamics. Their target application is face pose estimation. Arnaud and Memin [2] use particle filters for tracking features in image sequences. However, they focused on the development of a variant of the particle filter method that is applicable to the feature tracking domain. Schulz et al. [3] adapted Bayesian filtering algorithms to track multiple people. Bruce and Gordon [4] used particle filters for motion tracking but use a more sophisticated motion model than a simple velocity model. Bennewitz et al. [5] used the Expectation-Maximization algorithm to build models of tracks in an indoor environment. The positions that comprise a learned motion track were then used as states in a HMM which can be used to estimate the positions of people in that environment. POMDPs have also been used to model the movement of humans for better planning of robot paths [6]. Avenel et al. [7] use non-linear filters to track curves in an image processing application. Limketkai et al. [8] show how particle filter parameters can be learned from labeled data. Rathi et al. [9] adapt particle filters for tracking objects with varying shape. Particle filters have been applied to contour tracking in images [10]. However, in these applications the particle filter is expected to make only short-term predictions. Kalman filters [11] and particle filters [12] have been demonstrated for cyclone tracking using as observations the eye detected from multiple satellite sources.

3. PARTICLE FILTERING

We first describe the conventional particle filtering formulation and then describe our modification to be used in cases where only a database of historical state sequences is available instead of an explicit state model.

3.1 Sampling Importance Resampling

Particle filtering is a Monte Carlo simulation of a Bayesian filter. The goal is to estimate the true (but hidden) state of a system from a sequence of observations conditioned on the state. Let x_t denote the true state of the system at time t and z_t the observation received from the system. To operate the Bayesian filter, it is assumed that the *state model* is given. The state model, $p(x_t|x_{t-1})$, defines the probability of the system existing in a particular state at time t given the system's state in the previous step. The distribution of the system state at the initial step ($t = 0$) is assumed to be available as $p(x_0)$. The observation model, $p(z_t|x_t)$, that defines the probability of receiving a particular observation given the system's state at that time is also available. The model estimation problem is then to calculate the posterior probability distribution$p(x_t|z_1, z_2, \dots, z_t)$.

Recursive Bayesian filter algorithms maintain an estimate of the posterior distribution for every time-step where observations are available. At every time-step, the posterior distribution estimate from the previous time-step is updated using the state model and then the observations are assimilated into the distribution estimate using the observation model. If the structure of the posterior distribution is known before hand (for instance if the state and observation models are Gaussian), then the posterior distribution estimate can be represented exactly using the distribution parameters. Particle filtering is used when this structure is not known. In this technique, the posterior distribution at any time-step is represented as a finite set of random samples (the "particles") drawn from the state space with an associated set of weights. The state and observations models are used to update the collection of particles and weights. Intuitively, the weight of a particle is increased in proportion to the likelihood of receiving the observation from the state represented by that particle. It can be shown that as the number of particles increases, the state estimate approaches the optimal Bayesian estimate [13]. The basic formulation of particle filtering suffers from the problem that the weights of all but one particle approach zero, thus increasing the variance of the estimate [13]. Particle filtering algorithms therefore implement resampling of the particles at every step to reduce this phenomenon. Below we summarize one common particle filtering algorithm that is based on importance sampling from the available state model itself.

Algorithm: Sampling Importance Resampling Particle Filtering

```
Initialization:
Set $x_0^{(i)}$ by sampling from $p(x_0)$.
    $w_t^{*(i)} = \frac{1}{N}, 1 \le i \le N$
Update at every time-step $t$:
    Extend particle to time-step $t$: Set $x_t^{(i)}$ by sampling from $p\left(x_t|x_{t-1}^{(i)}\right), i = 1,2,\dots,N$
    $w_t^{*(i)} = w_{t-1}^{(i)} p\left(z_t|x_t^{(i)}\right)$
    Normalize weights: $w_t^{*(i)} \leftarrow w_t^{*(i)} / \sum_{i=1}^{N} w_t^{*(i)}$
    Resampling: Set $x_t^{(i)}$ by sampling with replacement from the set $\left\{x_t^{*(i)}\right\}$ with probabilities $w_t^{*(i)}$
```

Here, $x_t^{(i)}$ denotes the *i*-th particle, and $w_t^{(i)}$ its weight, $1 \le i \le N$. We next describe a modification of this algorithm that can be used when the state model, $p(x_t|x_{t-1})$, is not explicitly given.

4. REPLACING THE STOCHASTIC MODEL WITH A DATABASE OF STATE SEQUENCES

We consider the case where a large set of state sequences of the system from prior instantiations is available. Let $y^j_{1:N_j}$ represent the *j*-th such sequence in the database (note that the length of different state sequences will in general not be equal). In addition, we are given a distance metric to compare any two state sequences, $d\left(y^j_{1:N_j}, y^k_{1:N_k}\right)$. The state space model as used in the particle filtering algorithm may be thought of as "spreading" the posterior estimate in state space while the observation model narrows the estimate based on the observations. We show below how the database and distance metric can be used to spread the estimate of the posterior distribution in lieu of an explicit state model. We assume that the probabilistic observation model is still available.

The state of the tracker is maintained by a set of "particles" where each particle corresponds to one of the complete state tracks in the database. Since, in general, the state tracks in the database may not be aligned in time (i.e., do not begin at the same point in time of the evolution of the physical phenomenon), each particle also maintains the index of the current index in the corresponding state track. The particles are initialized by selecting randomly from all the tracks in the database. As in conventional particle filtering, a weight is maintained along with each particle. The weighted mean of the tracks corresponding to each particle represents the best estimated of the tracked phenomenon. With every observation, the following steps are performed:

1. Extend each particle's index into the state track by one
2. Compute distance between every particle and its corresponding state track in database
3. Resampling: replace farthest p_R% of the particles by sampling from the tracks in the database with weights proportional to the distance between the observation and the track.

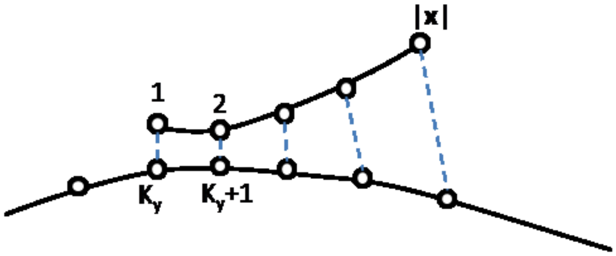

Figure 1: Defining the distance between two tracks as the mean distance between corresponding points.

Comparing this algorithm with the sampling importance resampling (SIR) particle filter, each particle is updated by just extending the index (whereas in the SIR particle filter, each particle is updated by sampling from the prior distribution estimate). In the SIR particle filter, the observation model is used to update the weights and the updated weights are used to resample the particles. In our algorithm, we replace a portion of the particles with tracks drawn from the entire database with probability proportional to the distance from the observation.

The algorithm relies on computing the distance between track segments (between observation segment and a state track, or between two complete tracks in the resampling step). We next define a distance metric for use in the tracking algorithm, followed by a detailed specification of the proposed algorithm.

4.1 Distance Metric

Let $\boldsymbol{x} = \{x_1, x_2, \dots, x_{|X|}\}$ denote a sequence of geospatial points. Here, $|x|$ denotes the length of the sequence $\boldsymbol{x}$. Given the set of points in $\boldsymbol{x}$, we can extrapolate the sequence to indexes outside the range $(1, |\boldsymbol{x}|)$. For simplicity, we assume that the points in $\boldsymbol{x}$ are uniformly spaced in time. If linear extrapolation is used, then the points on the extended sequence can be defined by

$$x_i^+ = \begin{cases} x_i, \text{if} 1 \le i \le |\boldsymbol{x}| \\ x_{|\boldsymbol{x}|} + (i - |\boldsymbol{x}|)\left(x_{|\boldsymbol{x}|} - x_{|\boldsymbol{x}|-1}\right), \text{if}\, i > |\boldsymbol{x}| \\ x_1 + (i-1)(x_2 - x_1), \text{if}\, i < 1 \end{cases}$$

We next define $d(\boldsymbol{x}, \boldsymbol{y})$, the distance between two tracks $\boldsymbol{x}$ and $\boldsymbol{y}$. The distance is defined as the mean distance between corresponding points on the two tracks. Since, in general, the two tracks will represent phenomena at different times the correspondence relationship between the points will have to be estimated. We assume that the first point on the first track and the point on the second track that is closest to it are corresponding points. As the tracks contain uniformly spaced points, the remaining points on the two tracks can be paired together in sequence. This is illustrated in Figure 1. Let K_y denote the index of the first corresponding pair of points:

$$K_y = \underset{1 \le j \le |\boldsymbol{y}|}{\operatorname{argmin}} \left\| x_1 - y_j \right\|$$

The distance, $d(\boldsymbol{x}, \boldsymbol{y})$, the distance between tracks $\boldsymbol{x}$ and $\boldsymbol{y}$ is then given by

$$d(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{|\boldsymbol{x}|} \sum_{i}^{|\boldsymbol{x}|} \left\| x_i - y^+_{i+K_y-1} \right\|^2$$

We note that a hurricane is a complex geospatial phenomenon and its size, intensity, and other properties evolve during its lifecycle. Ideally, such thematic properties of geospatial phenomena should be included while formulating the distance metric. Here, we have studied a formulation that only considers the eye locations.

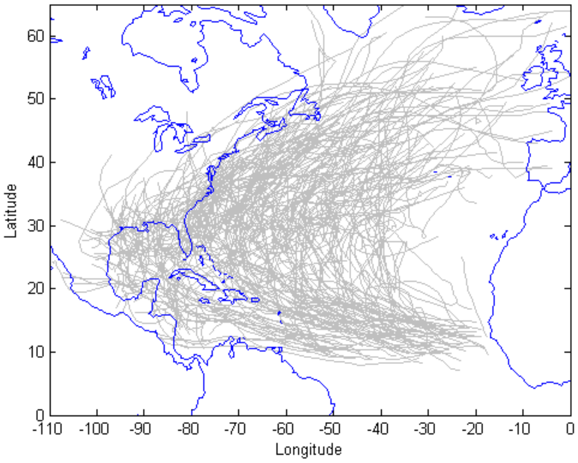

Figure 2: Paths of tropical cyclones in the Atlantic Ocean between 1980 and 2004.

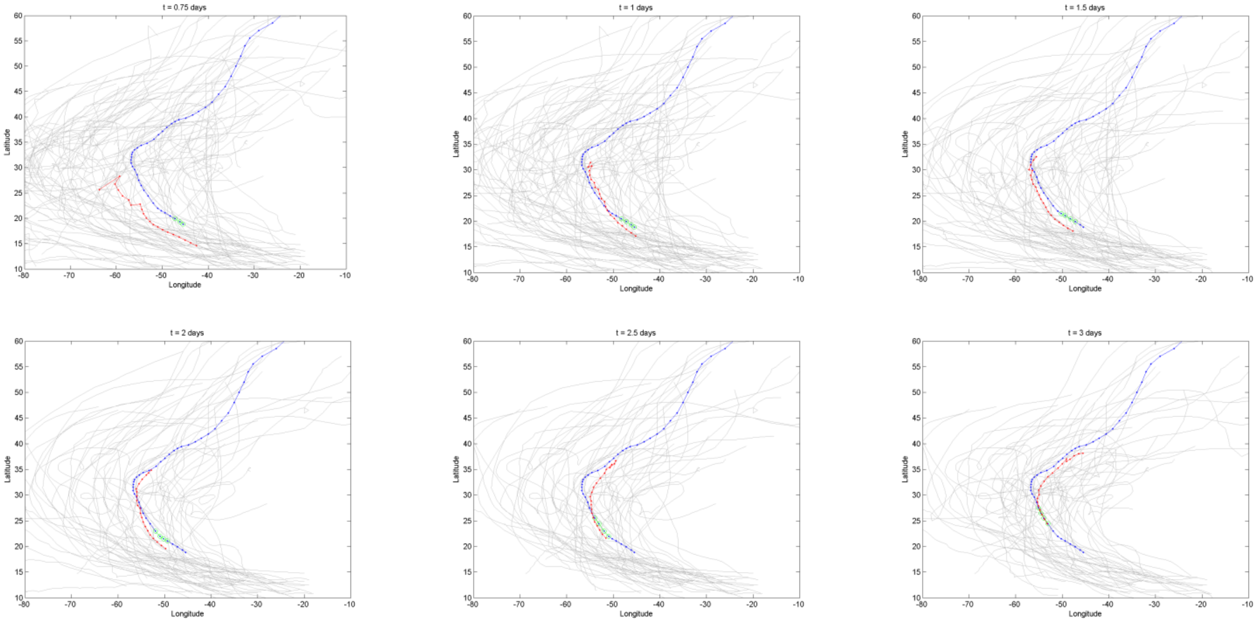

Figure 3: Predicting the paths of Hurricane Maria at different times. The blue line indicates the ground truth (NHC track). The green circles indicate the sequence of 4 eye locations that provide the observation. The red line indicates the predicted path of the cyclone. The gray lines denote the tracks in the database that correspond to the particles at that time.

4.2 Description of the Algorithm

Let N denote the number of particles and let D denote the set of complete state sequences (the database). Let the complete state histories in D be denoted by $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_{|D|}\}$ and $\boldsymbol{z}_k = (z_{k,1}, z_{k,2}, \dots, z_{k,L_k})$. $|D|$ is the number of unique state histories in D and $L_k$is the length of the k-th track in the database. Each particle corresponds to a complete state sequence and an index into that sequence. Denote the i-th particle at time-step t by the tuple $< M_t^{(i)}, K_t^{(i)} >$. Here, $M_t^{(i)}$ denotes the index of the state history in the database and $K_t^{(i)}$ indicates the offset into the track $\boldsymbol{z}_{M_t^{(i)}}$. Thus, the state sequence of particle i is given by $\boldsymbol{x}_t^{(i)} = \left\{z_{M_t^{(i)},K_t^{(i)}}, z_{M_t^{(i)},K_t^{(i)}+1}, \dots, z_{M_t^{(i)},L_{M_t^{(i)}}}\right\}$. p_R denotes the proportion of particles to be replaced with tracks close to the observed segment at every time-step.

Algorithm: Data-centric Particle Filtering

```
Input:
Set of observations $\{\boldsymbol{y}_t\}, 1 \le t \le T$
Initialization:
    Set $M_0^{(i)}$by randomly selecting from $\{1,2,\dots,|D|\}$; Set $K_0^{(i)}$ to
    a random number between 1 and $L_{M_0^{(i)}}$.
For every time-step $t$:
    $L = \emptyset$
    For every particle $i$:
        /* Extend every particle $i$ by incrementing index */
        if $K_{t-1}^{(i)} + 1 \le \left|\boldsymbol{x}_{t-1}^{(i)}\right|$
            $K_t^{(i)} = K_{t-1}^{(i)} + 1$
            $M_t^{(i)} = M_{t-1}^{(i)}$
        else /* delete particle $i$ */
            $L \leftarrow L \cup \{i\}$
    /* Resample a proportion of the particles */
    $d_t^i = d(\boldsymbol{y}_t, \boldsymbol{x}_t^{(i)})$
    Let $d^*$ be the $\lfloor p_R N \rfloor$ largest distance in $d_t^i, 1 \le i \le N$
    $L \leftarrow L \cup \{i\}, \forall d_t^i > d^*$
    $w_j = K \exp\left(-d\left(\boldsymbol{y}_t, \boldsymbol{z}_j\right)\right), \boldsymbol{z}_j \in D$
    Set $M_t^{(i)}, i \in L$ by sampling with replacement from
    $\{1,2,\dots,|D|\}$ with probabilities $w_i$.    □
```

At every time-step, L indicates the particles that will be replaced either because incrementing the track index exceeds the length of the corresponding state sequence or because the distance from the particle to the observation segment is too far (i.e., is one of the farthest $\lfloor p_R N \rfloor$ particles). These particles are replaced by random sampling from the database with weights (w_j) that are inversely proportional to the distance between a track in the database and the observation segment.

Note that since an explicit observation model is not available, resampling based on a new observation utilizes the distance metric between the observation and tracks in the database. This ensures that particles that are farthest from the observation are replaced with tracks from the database that are closest to the observation.

Given the state sequences of all the particles at a time-step and the relative weights, the state estimate at that time is defined as the weighted mean of the individual state trajectories of the particles.

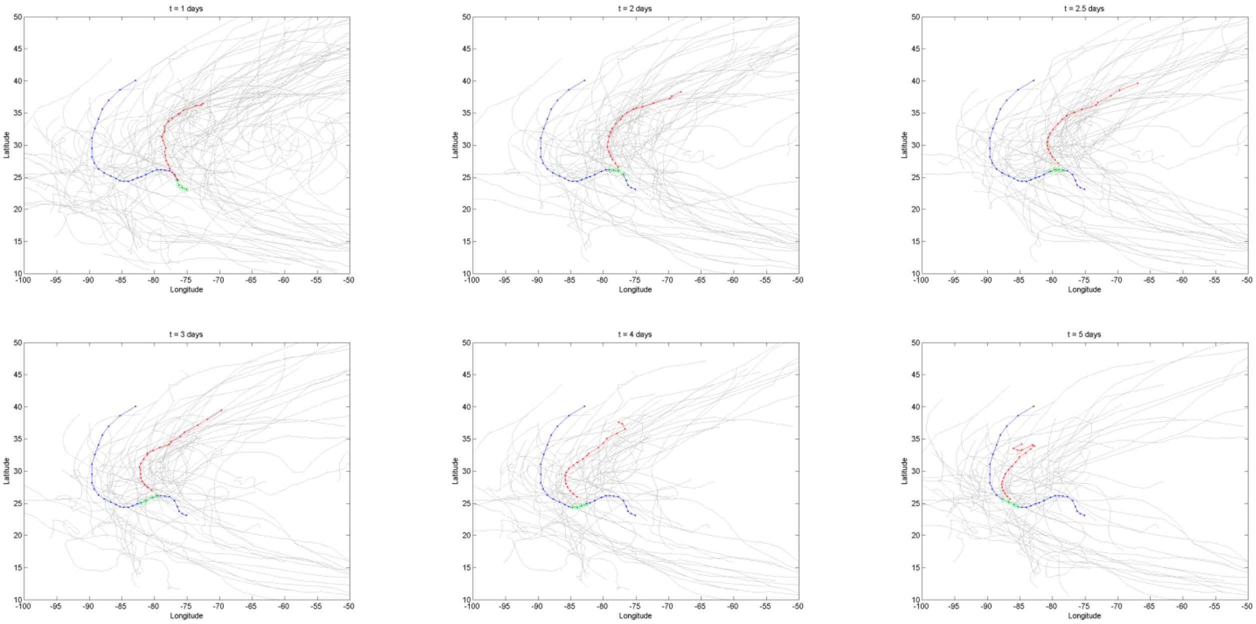

Figure 4: Predicting the paths of Hurricane Katrina at different times. The blue line indicates the ground truth (NHC track). The green circles indicate the sequence of 4 eye locations that provide the observation. The red line indicates the predicted path of the cyclone. The gray lines denote the tracks in the database that correspond to the particles at that time.

$$\langle \boldsymbol{x} \rangle_t = \frac{\sum_{i=1}^{N} \boldsymbol{x}_t^{(i)} w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}$$

For complex geospatial phenomena, the weighting should consider thematic properties (such as intensity of hurricanes). In this work, we have studied a weighting that is only dependent on the distance metric.

4.3 Run-time of the Algorithm

The runtime complexity of the algorithm can be computed for every time-step. This is dominated by the time taken to compute the weights for each particle and for each track in the database (for particle resampling). This time is in turn proportional to the time taken to compute the distance function between the observation segment and the particle or database track. Let T denote the time complexity to calculate the distance between two tracks. The time taken to compute the weights for all particles in the replacement set $L$is $O(|L|T) = O(NT)$ and the time taken to compute the weights of all tracks in the database for resampling is $O(|D|T)$. Let L_{MAX} denote the maximum length of a track in the database. Since the length of each observation is constant, $T = O(L_{MAX})$. Thus, the overall runtime complexity at each iteration is $O\big((N + |D|)L_{MAX}\big)$.

In the current formulation of the algorithm, all historical tracks available in the database are considered. The computation cost may become too expensive if the size of the database is large. In such cases, spatial, temporal, or thematic sorting may be performed before-hand to reduce the number of tracks that are included in the prediction.

5. DESCRIPTION OF THE DATA

To evaluate the proposed tracking algorithm, we will utilize the algorithm to predict the tracks of the eye of tropical cyclones. Tropical cyclones are low-pressure weather systems that develop over warm tropical waters of the ocean and are characterized by strong wind circulation about a well-defined center (the “eye” of the cyclone). In the United States, the National Oceanic and Atmospheric Administration’s National Hurricane Center (NOAA-NHC) is responsible for tracking tropical cyclones in the Northeastern Pacific Ocean and Northern Atlantic Ocean. The tracking task is manually performed using data from satellites, radar, reconnaissance aircraft, and ships, and surface observations from land stations and data buoys. Cyclone forecasting is based on numerical models and these tracking data.

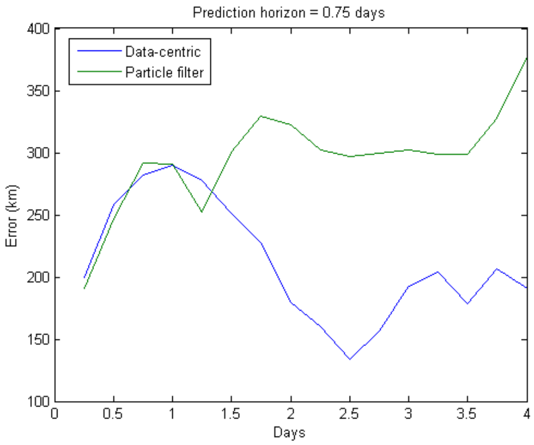

Figure 5: Prediction error (distance between the predicted and true cyclone eye locations) when using the proposed data-centric approach and conventional particle filtering. The prediction horizon is 18 hours beyond the last observation.

We used the tracks of all tropical cyclones (as reported by the NHC) that formed in the Atlantic Ocean between 1980 and 2004

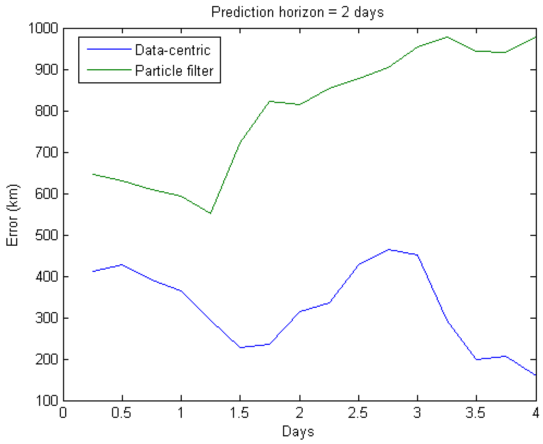

Figure 6: Prediction error (distance between the predicted and true cyclone eye locations) when using the proposed data-centric approach and conventional particle filtering. The prediction horizon is 48 hours beyond the last observation.

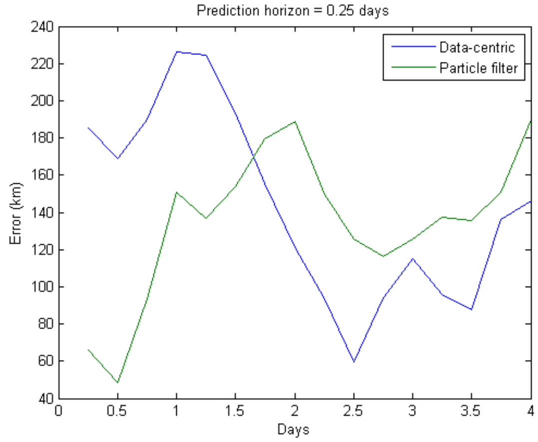

Figure 7: Prediction error (distance between the predicted and true cyclone eye locations) when using the proposed data-centric approach and conventional particle filtering. The prediction horizon is 6 hours beyond the last observation.

as the database of geospatial tracks. The NHC best-track is available at fixed 6-hour intervals. There are 276 cyclone tracks in the database. These are shown in Figure 2.

Storm eye locations from two cyclones in the 2005 Atlantic Ocean were used as observations. The particle filter-based algorithm was used to generate the estimated cyclone track of these 2005 cyclones given the 25-year database of cyclone tracks and the initial sequence of observations from the 2005 cyclones. We used the distance between the predicted cyclone eye location as output by the tracking algorithms and the cyclone eye location provided by the NHC ("best track") as a metric for objectively evaluating the tracking algorithms.

6. RESULTS

We tested the tracking algorithm on the cyclone tracks from 2005, the most active Atlantic hurricane season so far. From each track, we presented sequences of 4 consecutive eye locations (the observations) to the tracking algorithm. The weighted state estimates given that observation provides the predicted cyclone track in the future. Note that since the underlying particles each represent a complete state history, the prediction can extend significantly into the future. We report the distance between the predicted eye at some point in the future with the true cyclone eye at that time (from the NHC track) as the error.

Hurricane Maria was the fourth major hurricane in the 2005 season. It lasted 5 days, and reached an intensity of Category 3. Figure 3 shows the results of predicting the cyclone track using the proposed algorithm at different stages of the hurricane. Note that the path of this cyclone curves sharply and this makes developing a general predictive model difficult. As can be seen from the figure, the proposed data-centric tracking algorithm is able to predict the change in the cyclone direction. As in the case of other particle filter algorithms, the particles take some time to converge to the observations (the particles were initialized by randomly sampling from the cyclone track database. In this experiment, there were 1000 particles.

Figure 4 shows the results of predicting the track of Hurricane Katrina. The predicted path matches less well compared to Hurricane Maria. This is because Hurricane Katrina was a Category 5 hurricane that made landfall and there are relatively few similar cyclones in the database.

6.1 Comparison with Particle Filtering

We compared the quality of prediction from the proposed algorithm with that from the conventional particle filter, formulated with a simple first order velocity model. The state consists of only the eye location and velocity.

$$\boldsymbol{x}_t = [x_{Lat}\ x_{Lon}\ \dot{x}_{Lat}\ \dot{x}_{Lon}]^T$$

$$\boldsymbol{x}_t = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{t-1} + w_t$$

This assumes that the time between state updates will be uniform. w_t corresponds to the noise in the state model and is modeled as zero-mean and normally distributed.

$$w_t \sim N\left(0, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}\right)$$

(We represent the latitude and longitude in degrees and so the velocity is in units of change in degrees per size hours. Representing the spherical latitude-longitude coordinate system using these matrices is an approximation but is sufficient for the short distances involved and because most of the hurricanes are confined to the tropics). The observations, $\boldsymbol{y}_t$, correspond to the instantaneous location.

$$\boldsymbol{y}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}_t + v_t$$

Here, v_t corresponds to the noise in the observations and is also modeled as zero-mean and normally distributed.

$$v_t \sim N\left(0, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

The number of particles is set to 1000. Figure 7 compares the prediction error between particle filtering and the data-centric algorithms when the prediction horizon is 6 hours (one time-step). At this short prediction horizon, the particle filter algorithm has a lower error at the beginning of the track. This is expected as the cyclone moves slowly and a simple physics-based model is sufficient to predict the immediate future location. The accuracy of the data-centric algorithm depends on the presence of similar tracks in the database. Since it is unlikely that there is an extremely close match, the short-term accuracy is low.

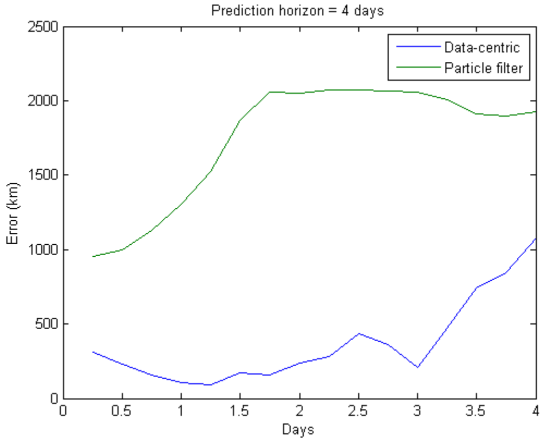

Figure 8: Prediction error (distance between the predicted and true cyclone eye locations) when using the proposed data-centric approach and conventional particle filtering. The prediction horizon is 96 hours beyond the last observation.

The accuracy of the data-centric approach improves relative to the particle filtering algorithm as the prediction horizon increases. This is shown in Figure 5, Figure 6, and Figure 8. The data-centric approach is able to predict the curve in the track from the presence of similar tracks in the database. On the other hand, the predictions from the particle filtering method are straight lines due to the first order model. Note that even with a prediction horizon of 4 days (Figure 8), the data-centric approach provides relatively accurate predictions, compared to that from the particle filter. Thus, the proposed algorithm is particularly suited for application where the accuracy of long-term predictions is important.

6.2 Effect of number of particles

The accuracy of particle filtering algorithms depends on the number of particles that are used to represent the state. We executed the data-centric tracking algorithm on the Hurricane Maria track using 200 and 1000 particles. The results are shown in Figure 9. There is no significant change in prediction accuracy with the different number of particles. Note that while there are only 276 unique tracks in the historic database, some of the 1000 particles index the same historic track but with a different starting offset.

An issue with the conventional particle filtering algorithms is the degeneracy problem – as the iterations increase, only one particle is assigned almost all of the weight, while the other particles have weights close to zero [13]. This increases the variance in the state estimate. One measure of the degeneracy, proposed in [13], is the effective number of particles at time-step t.

$$N_{eff}(t) = \frac{1}{\sum_{i=1}^{N} \left[w_t^{(i)}\right]^2}$$

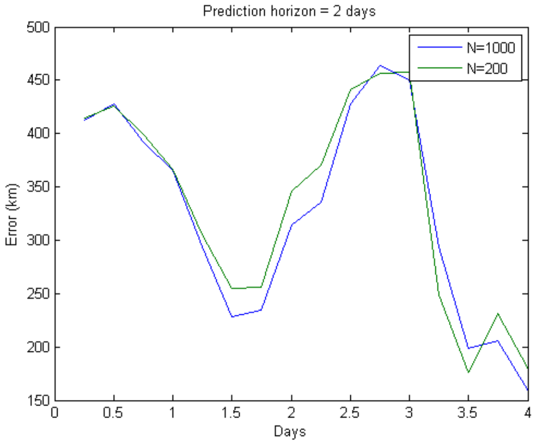

Figure 9: Prediction error (distance between the predicted and true cyclone eye locations) using the data-centric approach with different number of particles (200 and 1000). The prediction horizon is 48 hours beyond the last observation

Here, $w_t^{(i)}$ is the normalized weight of the i-th particle at time t and N is the number of particles. The effective number of particles computed for the data-centric algorithm, plotted along the track time, is shown in Figure 10. With repeated replacement of the particles with tracks selected from the database based on their distance from the observation, the effective number of particles approaches the theoretical maximum.

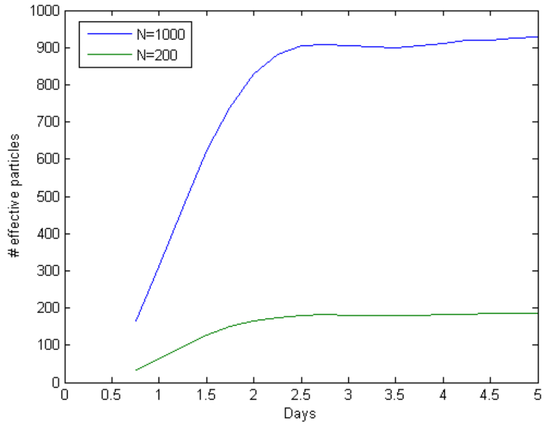

Figure 10: Effective number of particles. The two curves are derived from experiments with different number of particles (200 and 1000).

We also plot the number of unique state histories represented by the particles in Figure 11. As more observations become available, the particles begin to converge on those state histories in the database that are most similar to the observed tracks. Thus, the number of unique state histories decreases over time. Note that there are only 276 distinct tracks in the database, and hence there will necessarily be repeated tracks when the number of particles is 1000.

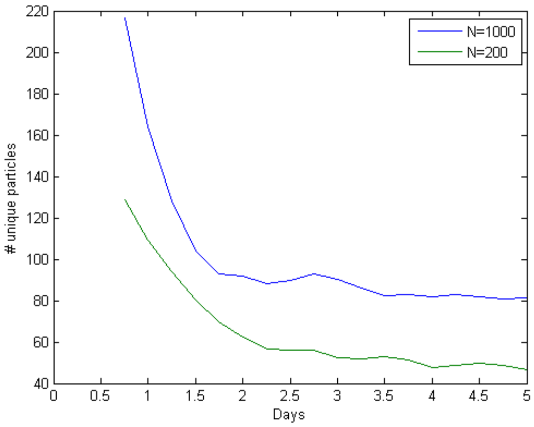

Figure 11: Number of unique state histories represented by the particles. The two lines are derived from experiments with different number of particles (200 and 1000).

7. CONCLUSION AND FUTURE WORK

We have described an algorithm for tracking geospatial phenomenon that is based on the concept of particle filtering. The algorithm is applicable instead of particle filtering if it is difficult to explicitly model the state transitions of the phenomenon but a database of historic state evolution trajectories is available. Thus, this algorithm is useful for tracking and predicting complex phenomena such as the movement of tropical cyclones. We have shown how the proposed algorithm performs for this application and compare the accuracy of predicting the future state with that of a particle filter formulated with a first order motion model. We found that while the short-term (6 hours) prediction accuracy is lower than that of particle filtering, the longer-term prediction accuracy is higher with the proposed method. Thus, the proposed algorithm is particularly suited for application where the accuracy of long-term predictions is important. We also showed results showing the effect of the number of particles on the tracking algorithm.

This data-centric approach to tracking complex spatial phenomenon has other applications besides cyclone tracking. For instance, modeling the movement of people in their every day environments is useful in applications in robotics and surveillance. Tracks in indoor spaces can be recorded using laser range-finders [14] while in outdoor spaces they can be recorded with handheld GPS receivers [15]. Such tracks often represent complex behaviors and are currently modeled by first learning stochastic models from previously observed tracks [16]. The proposed data-centric algorithm may be applicable in such applications as a repository of past tracks is available.

8. ACKNOWLEDGMENTS

This work has been sponsored by the NASA under an Applied Information Science Research Program (AISR) grant.

9. REFERENCES

[1] D. Mikami, K. Otsuka, and J. Yamato, "Memory-based Particle Filter for face pose tracking robust under complex dynamics," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 999 - 1006.

[2] E. Arnaud and E. Memin, "Partial Linear Gaussian Models for Tracking in Image Sequences Using Sequential Monte Carlo Methods," *International Journal of Computer Vision,* vol. 74, pp. 75 - 102, 2007.

[3] D. Schulz, W. Burgrad, D. Fox, and A. B. Cremers, "People Tracking with a Mobile Robot Using Sample-Based Joint Probabilistic Data Association Filters," *Intenational Journal of Robotics Research,* vol. 22, pp. 99-116, 2003.

[4] A. Bruce and G. Gordon, "Better Motion Prediction for People-tracking," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, New Orleans, USA, 2004.

[5] M. Bennewitz, W. Burgard, and S. Thrun, "Learning Motion Patterns of Persons for Mobile Service Robots," in *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002, pp. 3601-3606.

[6] M. Emaduddin and D. A. Shell, "Estimation of Pedestrian Distribution in Indoor Environments using Multiple Pedestrian Tracking," in *Workshop on People Detection and Tracking, IEEE Intl. conf. on Robotics and Automation*, 2009.

[7] C. Avenel, E. Mémin, and P. Pérez, "Tracking closed curves with non-linear stochastic filters," in *Conference on Scale Space and Variational Method*, 2009.

[8] B. Limketkai, D. Fox, and L. Liao, "CRF-Filters: Discriminative Particle Filters for Sequential State Estimation," in *International Conference on Robotics and Automation (ICRA)*, 2007.

[9] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking Deforming Objects Using Particle Filtering for Geometric Active Contours," *Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, pp. 1470-1475, 2007.

[10] P. Li, T. Zhang, and A. E. C. Pece, "Visual contour tracking based on particle filters," *Image and Vision Computing,* vol. 21, pp. 111-123, 2003.

[11] S.-S. Ho and A. Talukder, "Automated cyclone discovery and tracking using knowledge sharing in multiple heterogeneous satellite data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, 2008, pp. 928-936.

[12] A. Panangadan, S.-S. Ho, and A. Talukder, "Cyclone tracking using multiple satellite image sources," in *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle, 2009.

[13] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing,* vol. 50, pp. 174-188, 2001.

[14] A. Panangadan, M. Mataric, and G. Sukhatme, "Tracking and Modeling of Human Activity using Laser Rangefinders," *International Journal of Social Robotics,* vol. 2, pp. 95-107, 2010.

[15] L. Liao, D. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines. ," *Artificial Intelligence,* vol. 171, pp. 311–331, 2007.

[16] D. J. Patterson, L. Liao, D. Fox, and H. Kautz, "Inferring High-Level Behavior from Low-Level Sensors," in *Fifth International Conference on Ubiquitous Computing (UBICOMP-03)*, 2003, pp. 73-89.