Tracking a Runner in Forest Park Using a Particle Filter

I. Introduction

The focus of this project is a runner on Leif Erikson trail in Portland, Oregon, starting at the NW Thurman entrance and only running towards the Germantown entrance. A runner is assumed to be in one of two behaviors, moving forward at a nearly constant velocity or stopped to rest. Position observations are evaluated using a GPS enabled watch, specifically the Garmin Forerunner 235. The GPS measurements for longitude and latitude are erroneous due to tree coverage and the estimated path is erroneous due to the slow and nonuniform sampling rate. The sampling rate is considered slow compared to the runner's velocity and the frequency of turns on the trail. For a seasoned and novice runner, average pace is the common measure of performance. Errors in position measurements can lead to errors in pace calculations, a point of frustration for many runners.

The objective of this project is to better estimate the position, in terms of longitude and latitude, using a particle filter and prior knowledge of the path and the runner's movement. Two different particle filter algorithms are evaluated, the sequential importance sampling with resampling (SISR) and the auxiliary particle filter [1] (see appendix for algorithms).

II. Process Model

The state vector consists of 5 elements, longitude (*lon*), latitude (*lat*), heading (θ), velocity (v), and average velocity (m). The process model is described as follows:

$$x_{n+1} = \begin{bmatrix} lon_{n+1} \\ lat_{n+1} \\ \theta_{n+1} \\ \max(v_{n+1},0) \\ m_{n+1} \end{bmatrix} = \begin{bmatrix} lon_n + v_n * \cos(\theta_n) \\ lat_n + v_n * \sin(\theta_n) \\ A(lon_n, lat_n) \\ \beta * m_n \\ \alpha * m_n + (1-\alpha) * v_n \end{bmatrix} + \begin{bmatrix} w_{lon} \\ w_{lat} \\ w_\theta \\ w_v \\ w_m \end{bmatrix}$$

Eq. 1. Process model for a runner moving northwest on Leif Erikson trail used in the sequential importance sampling and auxiliary particle filters

Velocity (v) is limited such that it cannot drop below zero. $A(lon_n, lat_n)$ is an interpolated estimate for the trail heading at point (lon_n, lat_n). Beta (β) is a binomial random variable with p (β = 1) = 0.9 and p (β = 0) = 0.1. Beta represents two behavior states of a runner, moving (β = 1) and stopped (β = 0). The average velocity (m) is parametrized by alpha (α=0.95), to generate a moving average that favors a long term average.

The noise processes (w) are zero-mean, independent, and normally distributed and are defined as:

$$w_n = \begin{bmatrix} w_{lon} \sim N(0,\sigma^2{}_{lon}) \\ w_{lat} \sim N(0,\sigma^2{}_{lat}) \\ w_\theta \sim N(0,\sigma^2{}_\theta) \\ w_v \sim N(0,\sigma^2{}_v) \\ w_m = 0 \end{bmatrix}$$

Eq. 2. Noise processes for a runner moving northwest on Leif Erikson trail used in the sequential importance sampling and auxiliary particle filters. Each random number is zero-mean, independent, and normally distributed. σ^2 represents the variance for each random distribution.

III. Measurement Model

GPS observations are exported as a gpx file from the Garmin Forerunner 235. The measurement model is defined below, where vincenty refers to the Vincenty Formula for geodesic distances using an ellipsoidal model of the earth's surface [2]-[3]. This formula translates longitude and latitude degrees to meters. The axis is also shifted such that the start of the trail is at (0,0).

$$y_n = \begin{bmatrix} vincenty(lon_n) \\ vincenty(lat_n) \end{bmatrix} + [u_n]$$
$$where\ u_n \sim N_2(0,\Sigma_u)$$

Eq. 3. Measurement model for a runner wearing a GPS enabled watch with observations of longitude and latitude in degrees. The Vincent formula converts degrees to meters. Σ_u is the covariance matrix for the noise process.

III. Initialization and Parameter Values

Initialization

Leif Erikson trail is characterized using data from the Trail Run Project [4], consisting of 316 waypoints with straight line connections between waypoints. As the best source for trail coordinates, this data is used for heading approximation, result comparison purposes, and considered "truth" despite its limited information and accuracy.

From the waypoints and corresponding direction, trail heading is approximated using a nearest neighbor interpolation. This approximation is shown in the figure below.

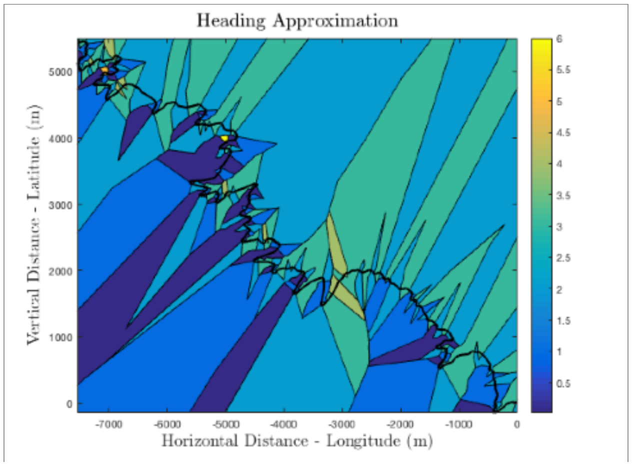

Fig. 1. Heading approximation for Leif Erikson trail using a neighbor neighbor algorithm for heading direction at each point. "True" heading values were calculated using waypoints from the Trail Run Project.

During a run, it is assumed that a runner will maintain a nearly constant velocity. However, the initial velocity was based on the average paces of 10,000 US runners over the course of a 5 kilometer race. The mean pace is approximately 11:45 min/mile (2.28 m/s). The probability starts to quickly decline around a 6 min/mile (4.5 m/s) and drops to zero at an elite sprinters pace of 12.5 m/s. Higher values are included with low probabilities to account for runners starting the race in a dead sprint. The probability distribution for initial pace is shown in the graph below.

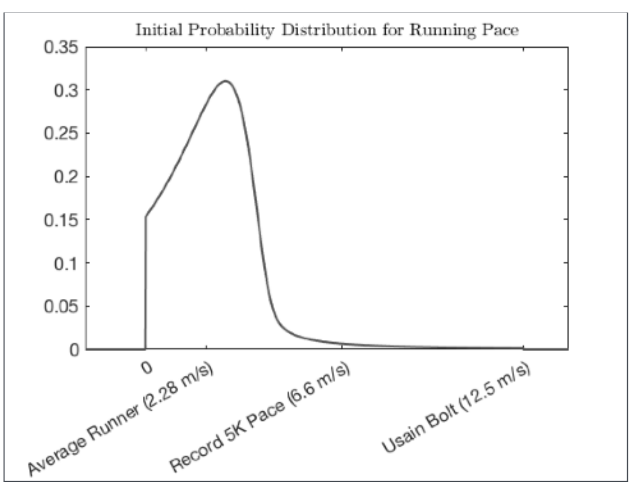

Fig. 2. Velocity (pace) initialization based on the distribution of 5K finishing times for 10,000 runners. The average pace is 11:45 min/mile and the probability quickly declines around 6 min/mile. Higher values are included with low probabilities to account for runners starting the race in a dead sprint.

Parameter Values

The process noises for each element are independent, zero mean, gaussian noise processes ($w \sim N(0,\sigma^2)$). Based on the trail width, longitude and latitude variance (σ^2) is 2 m^2. Heading variance (σ^2) is 0.75 rad^2, based on the turn angles seen on the trail and accounts for errors in the heading approximation. The velocity noise variance (σ^2) is 0.9 m^2/s^2, based on personal running data. Finally, the additive noise for average velocity is zero.

Measurement noise is an additive white gaussian process, ($u \sim N(0, \sigma^2)$) where σ^2 is 9 m^2, based on testing of the Garmin Forerunner 235.

Filter

Both algorithms were evaluated using 1000 particles. The resampling threshold for the SISR filter was set to 10%, or 100 particles. Additionally, the importance density was set equal to the prior. Estimates are derived from the mean of the posterior.

IV. Synthetic Simulation

Evaluating the model with synthetic data shows promising results. Generating 50 independent simulations each with 100 time steps and 1000 particles shows a mean square error of ~9 m^2. For comparison, the MSE between the synthetic measurement and true state is ~18 m^2. The graph below shows the MSE for 50 realizations. The two filters showed an improvement over the raw measurement, however there is no discernible performance difference between the SISR and the Auxiliary algorithms.

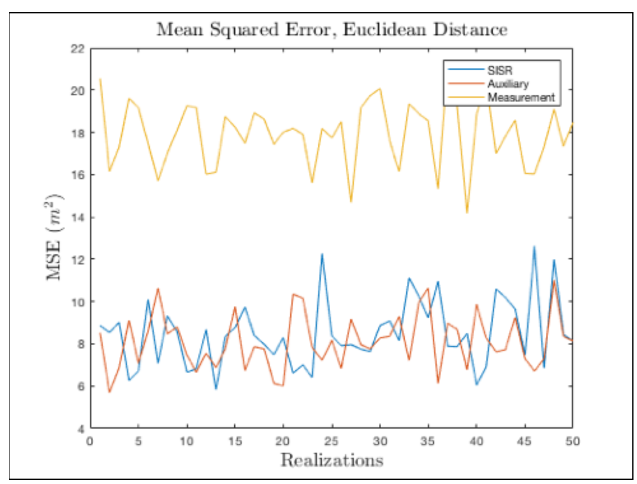

Fig. 3. Performance evaluation of the SISR and auxiliary particle filters in terms of mean squared error. Both filters show an improvement over the measurement (yellow); however, there is no noticeable performance difference between the two algorithms (blue and red).

Below is a true path and the measured (observed) path resulting from one realization.

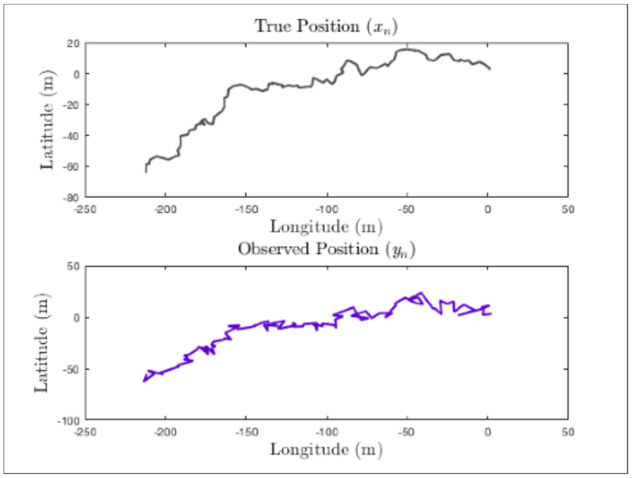

Fig. 4. Resulting true and observed states for a single synthetic realization of 100 time steps.

The posterior PDF for longitude and latitude are both unimodal and demonstrate a narrow region of high confidence. The posterior for heading shows a unimodal distribution with a wide region, indicating limited confidence. Finally, the posterior for velocity shows a bimodal distribution, with high probability between 1.5 and 4.5 m/s as well as around 0 m/s. This is as expected from the incorporation of the two behavior modes for velocity.

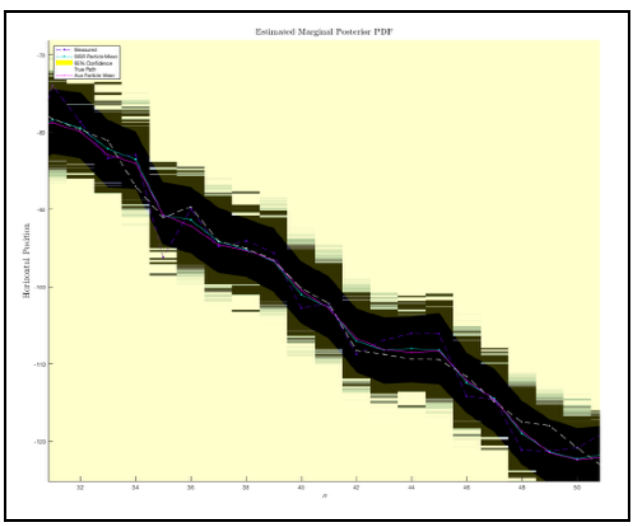

Fig. 5. Pseudo-colored representation of the marginal posterior probability for longitude.

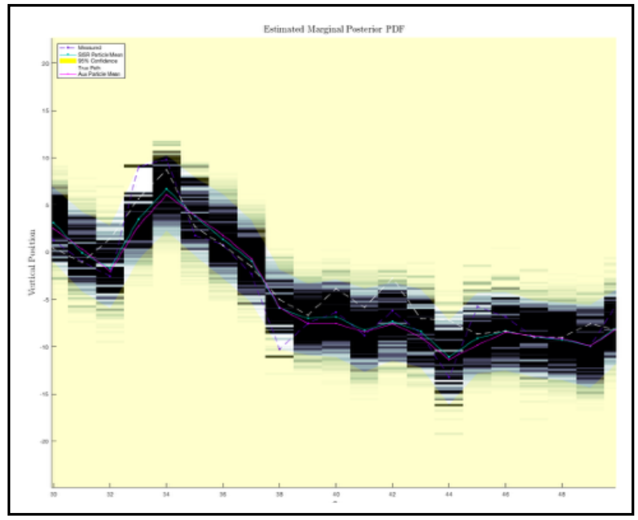

Fig. 6. Pseudo-colored representation of the marginal posterior probability for latitude.

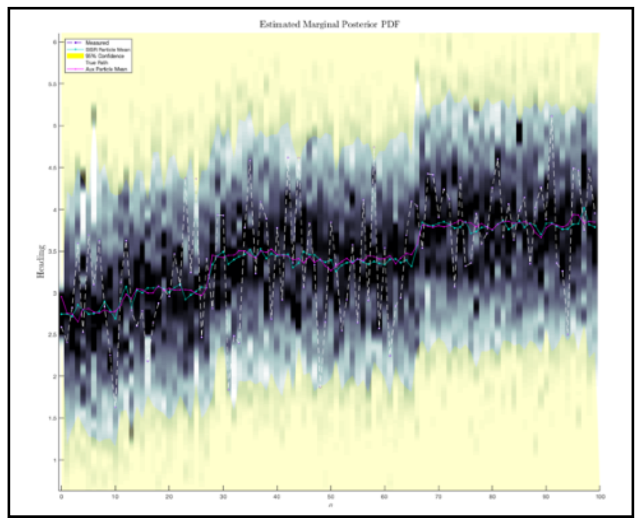

Fig. 7. Pseudo-colored representation of the marginal posterior probability for heading.

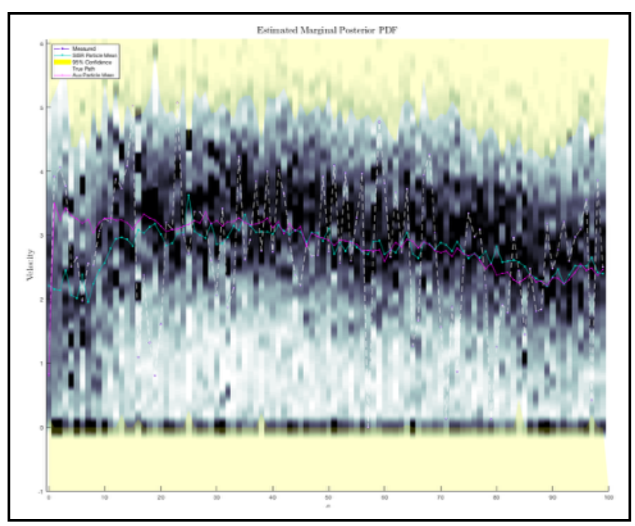

Fig. 8. Pseudo-colored representation of the marginal posterior probability for velocity.

IV. Testing

After evaluating the performance using synthetic simulations, each algorithm is evaluated using real running data. The testing dataset consists of 936 data points from the Garmin Forerunner 235, traversing the entire length of the Leif Erikson trail. The data has irregular time intervals ranging from 1 second between samples to 30 seconds between samples. Focusing in on a smaller region, specifically two regions where long periods of time pass with no measurement data, allows for closer evaluation of each algorithm. The SISR shows a nice track for wider turns in an area with no measurement data, as seen in the lower right corner the plot below, just after the star labeled 3000 (indicates time in sec). However, this algorithm breaks down with some tight turns due to the inaccuracies in the heading approximation as seen around point (-4700,2800). Additionally, near the center of the plot, around the star marked 4000, the SISR does not effectively estimate the stopped behavior.

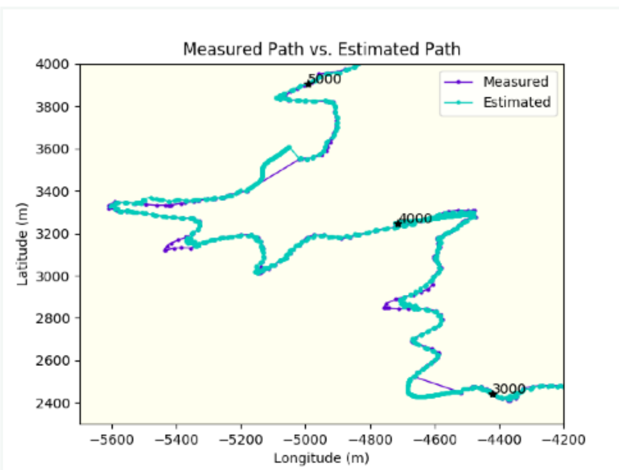

Fig. 9. Measurement data (purple) and resulting position estimate (teal) of the SISR particle filter. The black stars indicate time in seconds for easier referencing.

The same areas are evaluated for the auxiliary particle filter. The first area, just after star 3000, shows the filter starting off in the right direction but is short, where the estimates do not reach the next position measurement within the time gap. However, this algorithm does a much better job of estimating the tight turn around the point (-4700,2800). Additionally, the auxiliary particle filter handles the stopped behavior seen around star 4000 much better.

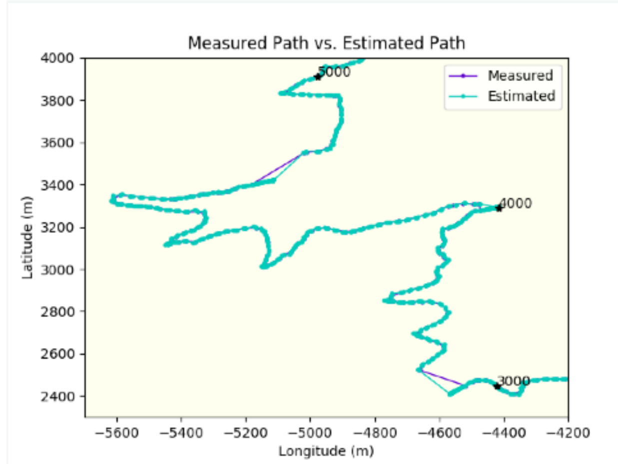

Fig. 10. Measurement data (purple) and resulting position estimate (teal) of the auxiliary particle filter. The black stars indicate time for easier referencing.

IV. Limitations

This model is designed for the Leif Erikson trail in Forest Park, Portland Oregon. Specifically it is designed for a runner starting at the NW Thurman trail head and only running one direction. Additionally, this model accounts for the runner stopping but not for turning around (negative velocities).

Another limitation arrises from the heading approximation. Given only 316 waypoints, the known trail does not accurately describe the turns in the park.

IV. Conclusion and Next Steps

By incorporating the prior knowledge of the trail and the runner, the SISR and auxiliary particle filters result in better position estimates compared to the raw measurements. Each have their own advantages and disadvantages; however, the first improvement needed is a more accurate and more specific approximation for the trail heading, requiring better trail coordinate data.

Additionally, given that the runner is reviewing this data after finishing the run, the viterbi filter might provide a better trajectory estimate by incorporating future measurements in the state estimation.

References

[1] O. Cappé S. J Godsill "An overview of existing methods and recent advances in sequential Monte Carlo" Proceedings of the IEEE vol. 95 no. 5 pp. 899-924 2007.

[2] Python GeoPy Library, https://geopy.readthedocs.io/en/1.10.0/#module-geopy.distance

[3] Vincent Formula, https://en.wikipedia.org/wiki/Vincenty%27s_formulae

[4] Trail Run Project, https://www.trailrunproject.com/trail/7001830/nw-leif-erikson-drive-trail

[5] Pace Calculator, http://www.pace-calculator.com/5k-pace-comparison.php

APPENDIX

Sequential importance sampling with resampling (SISR) algorithm from Cappe 2007 [1].

Algorithm 3 Particle Filter

for $i = 1, \ldots, N$ **do** ▷ Initialisation

Sample $\tilde{x}_0^{(i)} \sim q_0(x_0|y_0)$.

Assign initial importance weights

$$\tilde{\omega}_0^{(i)} = \frac{g(y_0|\tilde{x}_0^{(i)})\pi_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)}|y_0)} .$$

end for

for $t = 1, \ldots, T$ **do**

if Resampling **then**

Select N particle indices $j_i \in \{1, \ldots, N\}$ according to weights

$$\{\omega_{t-1}^{(j)}\}_{1 \le j \le N} .$$

Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(j_i)}$, and $\omega_{t-1}^{(i)} = 1/N$, $i = 1, \ldots, N$.

else

Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(i)}$, $i = 1, \ldots, N$.

end if

for $i = 1, \ldots, N$ **do**

Propagate:

$$\tilde{x}_t^{(i)} \sim q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t) .$$

Compute weight:

$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \frac{g(y_t|\tilde{x}_t^{(i)}) f(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})}{q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t)} .$$

end for

Normalise weights:

$$\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_t^{(j)} , \quad i = 1, \ldots, N.$$

end for

Auxiliary algorithm from Cappe 2007 [1].

Algorithm 4 Auxiliary Particle Filter

for $i = 1, \ldots, N$ **do** ▷ Initialisation

Sample $\tilde{x}_0^{(i)} \sim q_0(x_0|y_0)$.

Assign initial importance weights

$$\tilde{\omega}_0^{(i)} = \frac{g(y_0|\tilde{x}_0^{(i)})\pi_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)}|y_0)} .$$

end for

for $t = 1, \ldots, T$ **do**

Select N particle indices $j_i \in \{1, \ldots, N\}$ according to weights

$$\{v_{t-1}^{(i)}\}_{1 \le i \le N} .$$

for $i = 1, \ldots, N$ **do**

Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(j_i)}$.

Set first stage weights:

$$u_{t-1}^{(i)} = \frac{\omega_{t-1}^{(j_i)}}{v_{t-1}^{(j_i)}} .$$

end for

for $i = 1, \ldots, N$ **do**

Propagate:

$$\tilde{x}_t^{(i)} \sim q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t) .$$

Compute weight:

$$\tilde{\omega}_t^{(i)} = u_{t-1}^{(i)} \frac{g(y_t|\tilde{x}_t^{(i)}) f(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})}{q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t)} .$$

end for

Normalise weights:

$$\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_t^{(j)} , \quad i = 1, \ldots, N.$$

end for