TERRAIN-AIDED LOCALIZATION USING FEATURE-BASED PARTICLE FILTERING

Sneha Kadetotad
Department of Electrical Engineering
Pennsylvania State University
University Park, PA 16802
sxk462@psu.edu

Pramod K. Vemulapalli
Department of Mechanical and Nuclear Engineering
Pennsylvania State University
University Park, PA 16802
pkv106@psu.edu

Sean N. Brennan
Department of Mechanical and Nuclear Engineering
Pennsylvania State University
University Park, PA 16802
sbrennan@psu.edu

Constantino Lagoa
Department of Electrical Engineering
Pennsylvania State University
University Park, PA 16802
lagoa@engr.psu.edu

ABSTRACT

The localization of vehicles on roadways without the use of a GPS has been of great interest in recent years and a number of solutions have been proposed for the same. The localization of vehicles has traditionally been divided by their solution approaches into two different categories: global localization which uses feature-vector matching, and local tracking which has been dealt by using techniques like Particle Filtering or Kalman Filtering. This paper proposes a unifying approach that combines the feature-based robustness of global search with the local tracking capabilities of a particle filter. Using feature vectors produced from pitch measurements from Interstate I-80 and US Route 220 in Pennsylvania, this work demonstrates wide area localization of a vehicle with the computational efficiency of local tracking.

I. INTRODUCTION

There has been tremendous interest in recent years to develop techniques to locate and track a vehicle on the roadway to not only enhance the safety and security of the driver, but also for developing collision-avoidance systems, driver-assist systems and autonomous vehicle control [1-2]. Currently the Global Positioning System (GPS) is the primary means of locating the position of a vehicle. However, GPS suffers from many issues including slow update rate, poor signal reception and fragility of the whole system due to accidental or intended satellite disruption [2].

The above issues with GPS systems have prompted researchers to explore other techniques to either augment or replace GPS in outage situations. Alternate localization methods typically utilize a map-based approach [3] in which the roadways of interest are initially mapped by collecting certain sensor data and processing this data to extract different parameters of interest. The map created in this process has a record of the parameter of interest and the locations corresponding to where these parameters were found. To perform localization, the vehicle must be outfitted with the sensor and be supplied with the map. The computer onboard the vehicle then correlates the sensor readings to the map to find its position. Different researchers have used different types of sensors and data types to solve this localization problem. For example, Zlot et.al [4] have used a LIDAR sensor and have extracted different types of statistical parameters from the LIDAR data to create a map. Schindler et al [5] have utilized vision sensors in tandem with SIFT features to localize themselves in an urban environment.

A novel idea was proposed in [6], where terrain data (roll and pitch values) were used in the localization process. The advantages of using terrain data is evident because, unlike LIDAR and vision sensors the inertial sensors are not affected by external conditions such rain, dust, fog, visibility etc.

The task of map-based localization by any of the above sensors can be broken down into the global localization problem and the local tracking problem. In case of global localization the vehicle has no prior information about its location and the algorithm proceeds to search through the whole map to finds its static position. In local tracking, the algorithm is initiated with a prior knowledge about the vehicle location and the vehicle has to track itself in the map as it is moving by locally minimizing the error between measured and mapped data. Typically the global localization methods utilize feature matching techniques that perform matches by using data structures such as KD-trees [7] or vocabulary trees [8] to quickly search through the map. In contrast, local tracking algorithms typically utilize model-based tracking methods such as Kalman filtering, unscented Kalman filtering [9] or particle filtering [6].

The hybrid localization algorithm proposed in this work attempts to merge the key ideas of global and local tracking algorithms in order to obtain the benefits of both. The remainder of this paper is organized as follows: Section 2 describes the previous work and techniques behind the particle filtering algorithm used in this study. Section 3 explains the process of generating feature vectors. Section 4 describes the hybrid tracking algorithm and Section 5 describes the advantages of the proposed method over previous methods. Section 6 describes the results obtained when the algorithm was tested for two different datasets, and finally section 7 presents the conclusions and mentions possible directions for future work.

II. PARTICLE FILTERING ALGORITHM

This research presented in this paper builds upon the work in [6], which previously showed the use of a particle filtering algorithm to locally track a vehicle. In this prior study, roll and pitch information was found to closely correspond to the vehicle's position. The algorithm used these correlations to find the position of a vehicle in a two-stage approach: the preprocessing phase and the online phase. In the preprocessing phase, pitch and/or roll data values were collected while driving on the roadway, and then stored on board the vehicle.

The main steps involved in the online phase are as follows: Initially the map was populated by particles that were randomly placed on it. While driving down the roadway pitch values were collected. The particles were made to progress forward on the map through a propagation step using the equation

$$X_p^k = X_p^{k-1} + dX + w \qquad (1)$$

Here X_p^k is the position of the p^{th} particle at the k^{th} time step, dX is the distance the vehicle travels between iterations as inferred from odometry, and w is Gaussian white noise of variance, Q, is the variance of the odometry measurement. The Gaussian noise added accounts for noise in odometry measurements and also maintains a degree of randomness in the location of the particles to prevent them from converging to any location too quickly. After the update step, the particles were weighted based on the degree of match between the pitch value just collected off the roadway and the pitch value corresponding to each particle from the map database. The weighting scheme used was of the form

$$q_i = \mu^{-1} * \exp\left(-0.5 * R_\theta^{-1} * \left(\theta_a - \theta_{p,i}\right)^2\right) \qquad (2)$$

where θ_a is the pitch measurement, $\theta_{p,i}$ is the i^{th} particle's pitch corresponding to its position in the terrain map, and μ is a normalizing factor equal to the sum of the particle weights, R_θ is the variance of the attitude measurement.

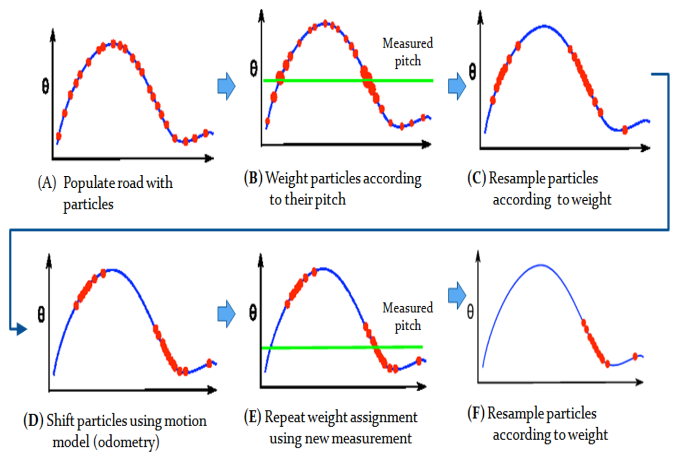

FIGURE 1. PARTICLE FILTERING TECHNIQUE [14]

After weighing each particle, a re-sampling step was conducted to eliminate particles with a low weight and multiply the ones with a high weight. The re-sampling step was basically a ranking scheme to give more importance to the particles which have a higher likelihood of being in the correct position. The details of the re-sampling step can be found in [14]. After re-sampling the particles, the position estimate was chosen to be a mean of the position of all the particles. This process was repeatedly performed to localize and then track the vehicle. Figure 1 illustrates the details of this process.

This approach to localization was validated on a wide variety of environments [14] such as highways and city roads. This shows that enough pitch variation is encountered on most roadways in order to perform localization. The localization ability of this method is not affected by the speed of travel of the vehicle as pitch is measured with respect to the travel distance. However, a vehicle travelling at a faster speed would localize quicker than a vehicle moving at a slower speed as the fast moving vehicle would encounter more variation in pitch in a given time interval. Recent work has also delved in to how the measurement noise in different pitch sensors affects the accuracy of localization [16].

The main disadvantage of the previous work [6] was that it uses all the pitch values collected and hence requires a vast database to store this information. It also requires considerable amount of computational effort to accurately localize the vehicle, quantified later in terms of FLOP Count (Floating-Point Operation Count).

III. FEATURE VECTORS

The previous work using inertial data for localization has typically utilized raw or filtered sensor data as an identifier to determine location [6]. In contrast, the present work draws upon the feature-based matching approaches [10] that have been built for global localization and utilizes these features that are extracted from raw sensor data as the parameters of interest. These features can be thought of as abstractions of the raw sensor data, abstractions that provide advantages in terms of uniqueness, computation and memory. This is critical because the localization algorithms will have to perform in a 'real-time' environment and also have reasonable memory requirements to be practicable. In the present work, feature vectors can be defined as unique information extracted from a signal that can be used to identify or locate the signal in a vast set of similar signals.

The feature vector generation process mainly consists of the following steps. First, a wavelet transform is performed to separate the data into different frequency bands. This is done because noisy data tends to dominate in the high frequency ranges. Then the signal corresponding to the lowest frequency band is chosen as the signal of interest. Figure 2 shows the process on an example signal. In this work a low-pass Gaussian filter was used with a cut-off frequency of .0074 cycles per meter, e.g. 1 cycle every 136 meters. This long spatial period was chosen as analysis of data showed that the long-period frequencies on a roadway gave the most repeatable features.

There are many ways to represent signals as features. One of the simplest is to use the maxima-minima points in a specific frequency band as the feature points. Each feature is represented by the value of the extrema points and the relative distance between the extrema point and its preceding extrema, as shown in figure 3. This use of relative distance instead of absolute distance makes the feature bias and scale-invariant, an important property because both errors are common in field data collection. Feature vectors are generated from pitch data using maxima-minima points, and every set of five such points forms a feature vector. The number of maxima-minima points chosen to form one feature vector, in this case being data from five extrema, depends on the degree of uniqueness desired. A very short feature would be less unique and hence give little advantage over raw pitch values, while a very long feature would require long time periods to detect on flat roadways and would hence cause a problem in quickly localizing the vehicle on certain sets of roads. The feature length of five points was chosen through tests on different road type datasets as a good tradeoff between uniqueness and convergence rates.

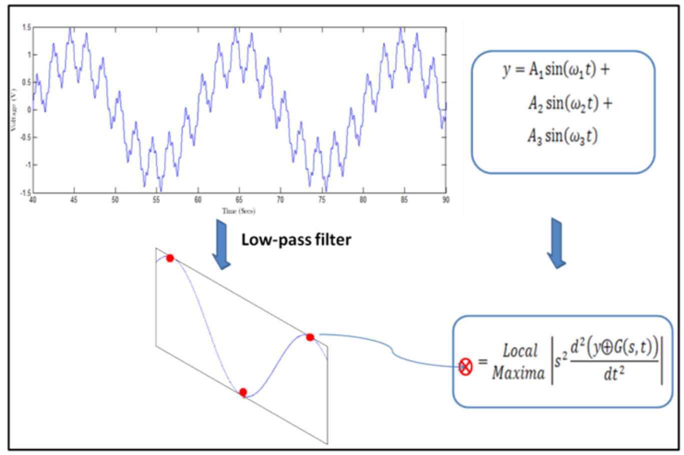

FIGURE 2. FEATURE DETECTION PROCESS [10]

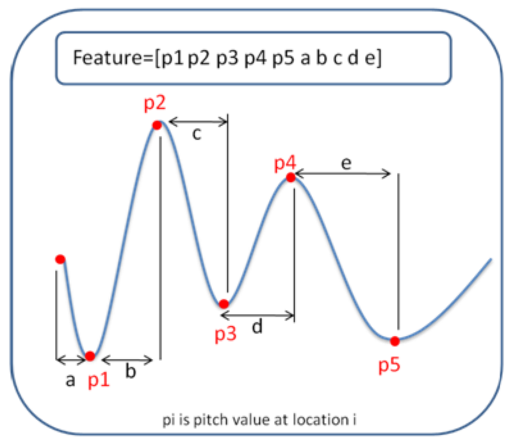

FIGURE 3. FEATURE VECTOR MEASURES

IV. FEATURE-BASED PARTICLE FILTER

The novel idea proposed here is to combine the concepts of particle filtering and feature generation to localize and track the vehicle efficiently. The particle filter can make use of the 'uniqueness' of a feature vector and hence go through the re-sampling step only when a feature is detected. This makes the algorithm more efficient and accurate by relying on the uniqueness of features rather than relying on every pitch value collected.

The process again consists of two main phases as described in Figure 4. In the preprocessing phase, pitch data is collected and correlated to specific locations on the roadway. Feature vectors are generated by the process described earlier. For each feature vector formed, the data stored in association with it are:

(1) The set of five pitch values
(2) Relative distance of each of these pitch values from its neighbors.
(3) Location of the last pitch value in that set.

Hence, each feature is storing information that correlates the pitch domain to the distance domain. This preprocessing phase is performed offline and the data is then stored on-board the vehicle.

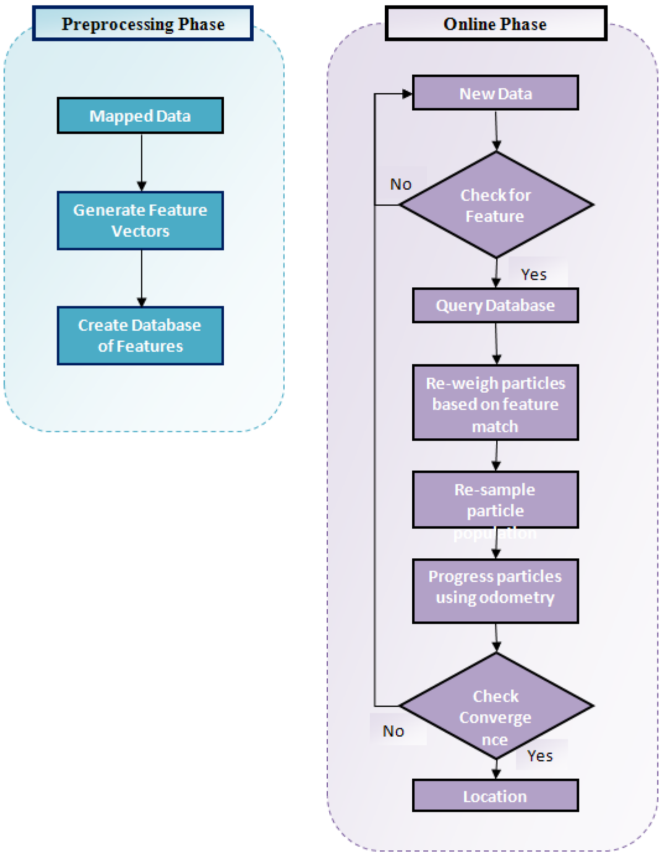

FIGURE 4. FEATURE-BASED PARTICLE FILTER

In the online phase, the localization process is initialized by populating the map with a set on N equally weighted particles randomly placed on the map, like a typical particle filter. As the vehicle drives down the roadway, pitch data is collected and the steps of wavelet transform and maxima-minima detection are performed as described earlier. As the vehicle keeps moving forward, the particles are also propagated in the map by a distance that is determined by odometry measurements, using equation (1) as in the regular particle filter. The main change introduced here is that, the particle filter correction step is not conducted unless a feature is detected; only then does it perform the re-weighting and re-sampling steps. The moment a set of five maxima-minima points are obtained, a feature is detected. Each particle located on the map is then re-weighted based on the degree of match between the feature just detected on the roadway and the feature 'associated' with the particle. The 'associated' feature for each particle is defined to be the feature most recently encountered by the particle from the map database. The weight given to the particle is based on two weights, a feature-matching weight and a distance-matching weight. The feature match-based weight is obtained from a Gaussian weighting function of the form in equation 2 with the difference that here instead of $(\theta_a - \theta_{p,i})^2$ the term used is $(\text{diff}(i))^2$ where

$$\text{diff}(i) = \|f(i) - f\|_2 \tag{3}$$

f(i) is the feature vector located closest to the i^{th} particle, but prior to the particle's position in the map, and f is the feature vector just detected in the online phase. In place of R_θ of equation (2), R_p, the measurement noise variance in pitch, is used. The norm used in equation (3) is the regular 2-norm of a vector.

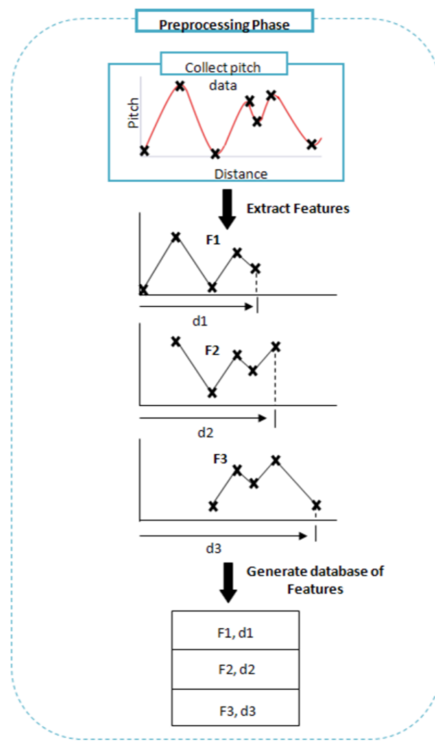

FIGURE 5. PREPROCESSING PHASE

The distance match-based weight is also a Gaussian weighting function of the form in equation (2) with

$$\text{diff}(i) = |\text{dist} - d(i)| \tag{4}$$

where d(i) is the absolute distance between the i^{th} particle and the feature prior to its position and d is the absolute distance between the feature just detected in the online phase and current position of the vehicle. In place of R_θ of equation (2), R_d, the measurement noise variance in odometry, is used.

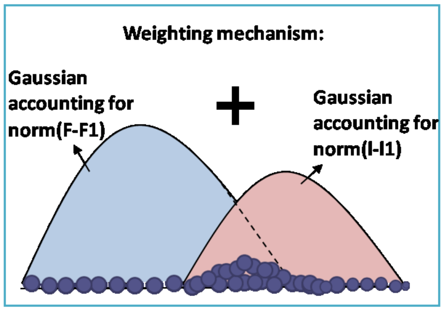

FIGURE 6. WEIGHTING MECHANISM FOR PARTICLES

These two weights are then combined as a weighted ratio to give a weight to each particle. Hence the weighting scheme used is as follows:

$$p_i^k = 0.8 * \left(Npf_i^k\right) + 0.2 * \left(Npd_i^k\right) \quad (5)$$

where
p_i^k is the weight given to i^{th} particle at the k^{th} iteration
Npf_i^k is normalized weight based on feature matching,
Npd_i^k is normalized weight based on distance matching.

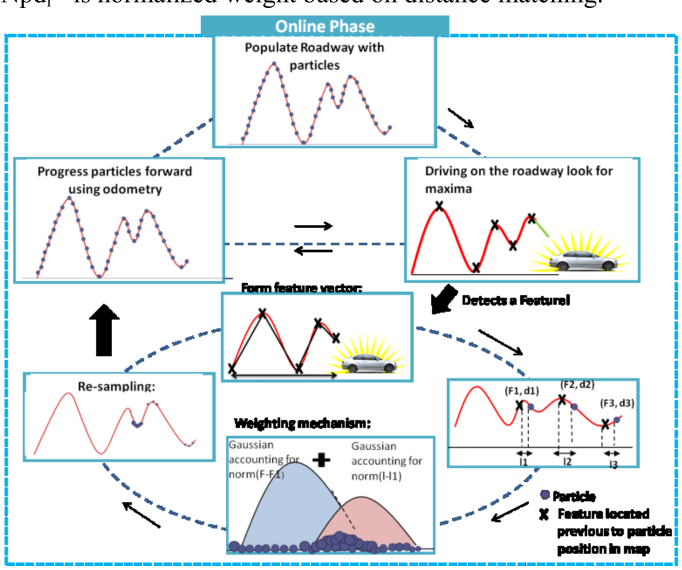

FIGURE 7. ONLINE PHASE

The ratio to combine the two weights was decided upon by a trial and error method and was found to give optimal results when the ratio of feature-matching weight to the distance-matching weight was 80/20. This ratio makes sense because a feature vector match holds more importance than a distance match as the probability of getting a good distance-match is more than a wrong feature-match. Hence, when a particle has a good feature match it will have a higher probability of being in the correct location.

The incorporation of distance-match ensures that there is no growing error in the position estimate. If not included, the position estimate would locate the position of the correct feature, but the location relative to the feature would be wrong since the vehicle has to drive past a feature before it can detect that feature.

Then the re-sampling step is performed as in a typical particle filter, as is shown in [6]. The above mentioned process continues as the vehicle drives down the roadway, resulting in eventual localization of the vehicle.

V. ADVANTAGES OVER PREVIOUS METHODS

The main idea of this paper is to show the feasibility of feature-based tracking in the context of vehicle localization. The eventual applicability of these methods in practical applications depends on the following underlying assumptions:

1) The road of travel has been previously mapped to obtain pitch values
2) On-vehicle storage of terrain information is possible.
3) Pitch data can be collected in a repeatable manner without considerable errors from the suspension dynamics, speed of travel etc.

The first two assumptions seem plausible due to ongoing research where large number of roadways are being mapped, and the increasing use and declining cost of on-vehicle data storage. The third assumption was substantiated in the research work conducted by Dean et al [14].

(1) Computational Effort

A very important aspect of comparison between algorithms is the computational effort that is involved. The regular particle filtering algorithm performs both update and re-sampling steps at each time step. Hence it was expected that the computational effort of this algorithm would be much higher than the feature-based particle filter, which performs the re-sampling step only when a feature is detected. A comparison of the number of computations involved in both algorithms was conducted, considering only one particle going through the entire process for all time steps for one dataset. As was expected, the number of FLOP counts in the regular particle filter was 3.06E8 while that in the feature-based particle filter was 3.52E7. The details of these calculations are described in table 1 below. The values for FLOP counts used were referenced from [15]. There is an order of magnitude difference between the computational effort involved for the two algorithms, the feature-based method being less computationally challenging. The huge decrease in computational effort is a very big advantage of the proposed method.

Another important point of comparison would be the number of particles required in each, as this greatly affects the computational burden. As features are more unique than pitch values, they have a higher chance of being a correct match when a match is obtained, as compared to pitch values. Hence the number of particles that need to be placed on the map for localizing the vehicle would be expected to be less in the case of a feature-based particle filter. The number of particles chosen in the regular particle filter (1000/mile) is explained in [6].

Tests were conducted using the feature-based particle filter and as expected, better localization accuracies were obtained when much smaller numbers of particles were used, as shown in the results section.

TABLE 1. COMPUTATIONAL EFFORT COMPARISON

OPERATION	FEATURE-BASED APPROACH	REGULAR PARTICLE FILTER
ADDITIONS	20,601,639	153,020,000
SUBTRACTIONS	825,293	4,003
MULTIPLICATIONS	22,983	6,000
DIVISIONS	3,447,375	38,260,000
EXPONENTIATIONS	2,151	2,000
TOTAL FLOP COUNTS	35,256,623	306,086,003

(2) **Global localization and local tracking**

Another important aspect to be considered is the fact that this approach is capable of performing global localization too. As explained in [10], the approach of using feature vectors generated from terrain information has already been shown to effectively perform global localization. Hence, even when the initial location of the vehicle is not known, this approach can be used to perform the global localization as well as the local tracking of the vehicle. As currently set, the algorithm performs both. Further, if the initial position is already known, then the local tracking of the vehicle can be performed with a much lower computational effort.

VI. RESULTS

The proposed algorithm was run on two different datasets collected from highways in the state of Pennsylvania. In both cases the feature-based particle filter requires a lower computational effort and gave better accuracy than the regular particle filter. This demonstrates the advantages of using the proposed method over the existing particle filtering technique. Figure 8 shows the position estimate error as a function of the distance travelled by the vehicle for the first dataset.

In the online phase, the regular particle filter reached a prediction error of less than 0.5m after 2800m of travel, while the feature-based particle filter reached a prediction error of below 0.5m after just 792m of travel. As seen in the Figure 8, convergence was reached at less than one-third the corresponding distance for the regular particle filter.

Also, the average error after convergence in the case of the regular particle filter was 0.7565m, while for the feature-based particle filter the average prediction error was 0.5984m. In the regular particle filter the number of particles placed on the map was 1000/mile, while in the case of the feature based particle filter they were 250/mile. Table 1 shows that there is 10 times less computational effort involved for the feature-based filter as compared to the regular particle filter. Hence, there is a 10x improvement in terms of computational effort given the same number of particles and a 4x improvement in terms of number of particles, giving an overall 40x improvement on the existing particle filtering technique.

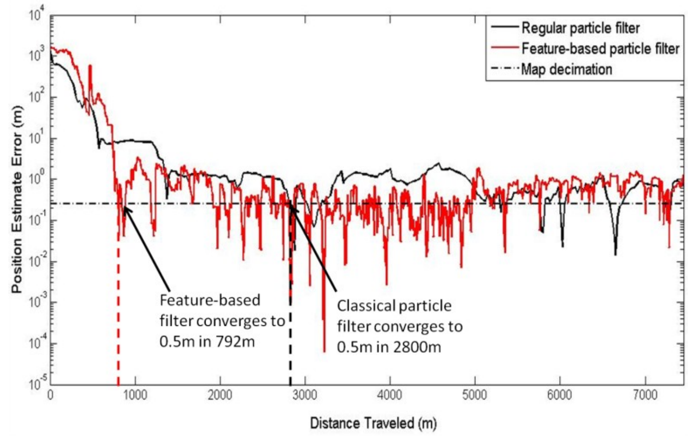

FIGURE 8. POSITION ESTIMATE ERROR VS DISTANCE TRAVELLED FOR DATASET 1

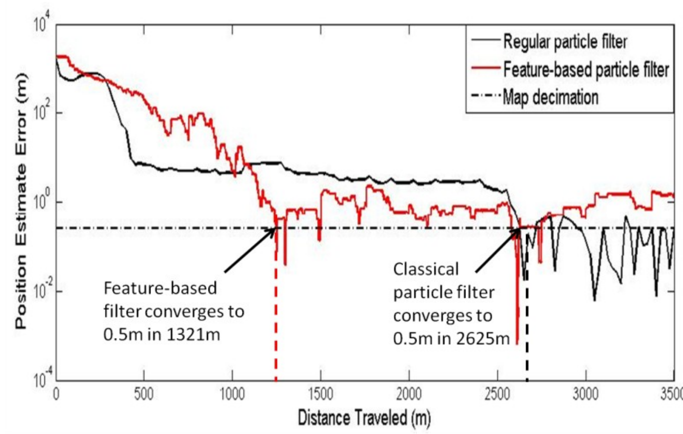

FIGURE 9. POSITION ESTIMATE ERROR VS DISTANCE TRAVELLED FOR DATASET 2

Figure 9 shows the results of running the online phase for both the algorithms on the second dataset. In the case of the regular particle filter, a prediction error of below 0.5m was reached after 2625m of travel while in the case of the feature-based particle filter a prediction error of below 0.5m was reached after 1321m of travel, almost half the distance. The average prediction error for the regular particle filter was 1.84m while for the feature-based particle filter the average prediction error was 0.84m The feature based approach worked with greater accuracy while using half the number of particles compared to a particle filter with raw data. The regular filter had 1000 particles/mile. In this case also the feature-based particle filter worked better with half the number of particles, 500 particles/mile. Hence in this case, there was an overall improvement of 20x as compared to the regular particle filter.

Another very important aspect of comparison is the database of information that needs to be stored in both the cases. The feature-based particle filter needs to store information pertaining only to the feature vectors whereas the regular particle filter needs to store all the pitch values collected off the roadway. Therefore, the database required is much smaller in the case of the feature-based particle filter. For simulation using the first dataset, the number of pitch values stored, in the preprocessing phase, for the regular particle filter was 4.10MB while using the same dataset for the feature-based particle filter required only 55.2KB, a reduction of a factor of 75. In the case of the second dataset, the regular particle filter needed a database of 2.46MB while the feature-based filter needed a database of just 6.63KB, an improvement by a factor of 380.

VII. CONCLUSIONS AND FUTURE WORK

The simulation results of the feature-based particle filtering approach show that, compared to a particle filter with raw data, a vehicle can be localized and tracked with a much higher convergence rate, with much better accuracy, through this method than using just the Particle filtering algorithm or the feature-based method. The algorithm works at a much lower computational effort and with a significantly smaller database, and thus is far more efficient

Future work can be directed towards improving on the feature-based approach by making the feature algorithm run parallel to the particle filter and help it whenever convergence is lost, as well as improve its efficiency by estimating bias and scale factor errors from the feature matches. Work can also be done in making the feature storage more efficient using approaches involving data structures such as KD-trees for performing the search.

REFERENCES

[1] J.C. Gerdes and E.J. Rossetter, “A Unified Approach to Driver Assistance Systems Based on Artificial Potential Fields.” ASME *Journal of Dynamic Systems, Measurement, and Control*, 123:431-438, September 2001.

[2] D.M. Bevley, J.C. Gerdes, C. Wilson, and G. Zhang, “The use of GPS-based velocity measurements for improved vehicle state estimation”, *Proc. American Control Conf.*, pp.2538-2542, 2000.

[3] K. Li, Han-Shue Tan, and J. Karl Hedrick., "Map-Aided GPS/INS Localization Using a Low-Order Constrained Unscented Kalman Filter." Shanghai : *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009.

[4] M. Bosse and R. Zlot, "Keypoint design and evaluation for place recognition in 2D lidar maps." *Zurich, Switzerland : s.n.*, 2008.

[5] G. Schindler, M. Brown and R. Szeliski, "City-scale location recognition." Anchorage, Alaska : *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[6] A. Dean, P. Vemulapalli and S. Brennan, "Highway Evaluation of Terrain-Aided Localization Using Particle Filters." Ann Arbor, Michigan : *ASME Dynamic Systems and Control Conference*, 2008.

[7] J. K. Uhlmann, “Algorithms for multiple-target tracking. *American Scientist, Comm. of the ACM*, 18(9), 1975.

[8] K. Mikolajczyk, H. Uemura, “Action recognition with motion-appearance vocabulary forest, *CVPR*, 2008.

[9] A. Dean, J. Langelaan, and S. Brennan, “Initializing An Unscented Kalman Filter Using a Particle Filter.” Proceedings of the 2009 *Dynamic Systems and Control Conference*, Hollywood, CA, USA, October 12-14, 2009.

[10] P. Vemulapalli, A. Dean, and S. Brennan, “Pitch-based Vehicle Localization using Time Series Subsequence Matching with Multi-scale Extrema Features”, To appear in *Proceedings of the 2011 American Control Conference*, June 29th –July 1st, San Francisco.

[11] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, 2002. “Particle filters for positioning, navigation, and tracking”. *Signal Processing, IEEE* Trans. on, 50(2), Feb., pp. 425–437.

[12] F. Caron, M. Davy, E. Duflos, and P. Vanheeghe, 2007. “Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning”. *Signal Processing, IEEE* Transactions on, 55, June, pp. 2703–2719.

[13] F. Chausse, J. Laneurit, and R. Chapuis, 2005. “Vehicle localization on a digital map using particles filtering”. In *Intelligent Vehicles Sym*., 2005. Proc. IEEE, pp. 243–248.

[14] A. Dean, “Terrain-based Road Vehicle Localization using Attitude Measurements”, Ph.D. Thesis, Mechanical Engineering, The Pennsylvania State University, October 2008, pp.17.

[15] R. Hockney, 1996. “The Science of Computer Benchmarking”, Society for Industrial and Applied Mathematics,3600 University City Science Center, Philadelphia, PA 19104-2688, Chap. 2, pp. 21-22.

[16] K. Jerath, and S. Brennan, “GPS- Free Terrain-Based Vehicle Tracking Performance as a Function of Inertial Sensor Characteristics.” to appear in the Proceedings of the 2011 *Dynamic Systems and Control Conference*, Arllington,VA, USA, October 31- November 2, 2011.