Pitch based Vehicle Localization using Time Series Subsequence Matching with Multi-scale Extrema Features

Pramod K. Vemulapalli*, *Student Member, IEEE*, Adam J. Dean and Sean N. Brennan, *Member, IEEE*

***Abstract*— Non-GPS localization of vehicles on roadways has received considerable attention in recent years and a number of solutions have been proposed, with most solutions addressing local tracking. This paper presents an algorithm that achieves global localization within very large road networks using pitch information. A key contribution is the development of the Multi-scale Extrema Feature that provides a number of advantages over traditional time-series subsequence matching methods in order to implement the above scheme. The algorithm's results in localizing a vehicle's position without initialization within a road network spanning 6000 Km are also presented.**

I. INTRODUCTION

Autonomous vehicles, driver-assist systems, collision warning systems, etc. all benefit from accurate estimates of vehicle location. While GPS provides position information, it is quite susceptible to attack, outages and signal reception problems. Consequently, there is growing interest to develop alternatives to GPS such as map-based localization techniques [1], including some which rely on onboard LIDAR [2] and vision sensors [3].

The problem of map-based localization can be broken down into two phases: global localization and local tracking. Global localization tries to estimate the position of the vehicle in the initial phase, during which it could be present anywhere in the map. Once the vehicle has been localized on a global scale, the second phase, i.e. local tracking, is initiated. In local tracking, the current position estimate of the vehicle must be determined from previous, nearby position estimates and current sensor information. Popular approaches to local tracking include Particle Filtering (PF) and variants of Kalman Filtering [4], both of which have been implemented by the authors for vehicle localization using pitch information [5] [6].

Global localization tends to be a harder problem to solve than local tracking because the large search space requires tremendous computational resources to implement a particle-filter or multiple Kalman Filter solution when the initial vehicle position is completely unknown. Previous approaches to global localization have involved the use of LIDAR and vision sensors [3; 4; 7; 2]. While both the above sensors are proven to improve the localization of a vehicle, they typically tend to fail under rainy and dusty conditions; moreover, vision sensors also tend to be unreliable during poor lighting conditions such as nighttime driving. These sensors are also expensive and can be blocked by dirt or snow. This work overcomes these problems by proposing the use of road grade data measured from in-vehicle INS sensors, which are robust under all the above conditions assuming vehicles are operating on known (e.g. mapped) roadways. Among the different global localization methods that have been suggested in literature [2] [3], the authors feel that the proposed method is most practicable for implementation on roadways as it is immune to external environmental conditions.

The data density per unit distance traveled also affects the sensor choice for localization. This choice is dependent on a trade-off between computation and travel distance for localization. The previously mentioned vision and/or LIDAR approaches utilize high densities of data per unit distance traveled, and hence require higher computational resources to localize, even within a small roadway network. At the same time their advantages include a smaller travel distance before localization, and the ability to simultaneously estimate multiple vehicle state parameters. In contrast, this paper utilizes pitch data which is one dimensional in nature. While pitch data requires only a moderate amount of computation and memory for localization within considerably large road networks, the tradeoff is that the vehicle must travel a larger distance before successful localization and only a single parameter (roadway location) can be estimated.

In order to implement a global localization scheme with pitch data, a signal retrieval method (time series subsequence matching) has been implemented. This approach is similar to recent approaches in vision and LIDAR-based global localization [2][3] which have utilized techniques from image retrieval [8] [7].

While previous work by the authors has focused on *local tracking* using pitch information, the main contribution of this paper is to demonstrate the feasibility of *global localization* in large roadway networks. This novel application of pitch information requires new time series subsequence matching tools because of the rather unique nature of this data. Therefore, the other major contribution of this paper is the 'Multi-Scale Extrema Feature' which is a feature vector that has been specially designed to facilitate

* Corresponding author

Pramod.K.Vemulapalli is with the Department of Mechanical Engineering, The Pennsylvania State University, State College, PA 16803 USA (phone: 814-321-2800; e-mail: pkv106@psu.edu).

Adam.J.Dean, was with Pennsylvania State University, State College, PA 16803 USA. He is now with the Department of Mechanical Engineering, Brigham Young University - Idaho, ID 83460 USA (e-mail: deana@byui.edu).

Sean.N.Brennan is with the Department of Mechanical Engineering, The Pennsylvania State University, State College, PA 16803 USA (phone: 814-863-2430; fax: 814-865-9693; e-mail: sbrennan@psu.edu).

pitch data retrieval from vehicles on roads. The feature based approach that has been developed for global localization could also be applied to local tracking problems [5; 6] to take advantage of the computational and memory benefits that are obtained in using this method.

The remainder of this paper presents an algorithm that achieves global localization within very large road networks, using pitch information. Section 2 presents the literature survey for current signal retrieval techniques and explains the need to develop a new feature vector that is particularly effective in handling the challenges presented by pitch data. This section also presents the proposed 'Multi-Scale Extrema Feature' and combines it with a KD-tree framework for global localization. Section 3 shows the algorithm's results in localizing a vehicle's position without initialization within a road network spanning 6000 km. Section 4 demonstrates the algorithm's immunity to typical types of sensor noise. Conclusions then summarize the main results of this work.

I. SCALED SHAPE FEATURES

A. *General overview of feature vector based localization*

In this paper, pitch data measured on a vehicle is compared to features stored in a map in order to search for the locations of maximum agreement. Implementation of the localization algorithm can be divided into two phases: a preprocessing phase and an online phase. In the preprocessing phase, mapped data is processed to obtain feature vectors which are stored in a database. In the online phase, data collected on the vehicle is used to create feature vectors which are used in conjunction with the database in order to obtain the location of the vehicle. A schematic that illustrates the above process is shown in Fig. 1.It is can be seen that the feature vector plays a critical role in both phases.

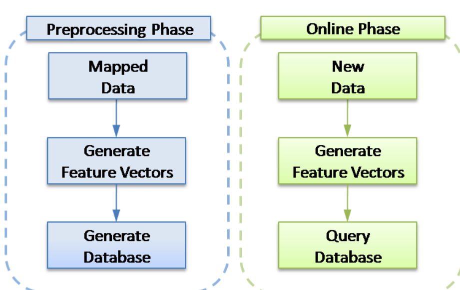

Fig. 1. The two phases involved in the proposed localization scheme and the central role played by the feature vector in these phases.

B. *Previous methods from the literature*

One can observe that this map-matching problem is quite analogous to the time series subsequence matching problem, for which a large number of different solutions have been proposed [9; 10; 11; 12] . Unfortunately, there are several drawbacks when applying these approaches to map-based localization. For example, all the above methods use the Euclidean distance or a modified Euclidean distance such as Dynamic Time Warping (DTW) to calculate the distance between the database signals and the query signal. The Euclidean distance criterion and its variants are ill suited for handling data with outliers or data in which the noise is present in particular frequency bands. Unfortunately, road information has larger noise content at higher frequencies to an extent where one would wish to create separate feature vectors from different frequency bands (or different scales) to isolate noise influences. These requirements are hard to meet with methods based on Euclidean criterion as one would need to maintain a separate matching process for each frequency band. Additionally, the Euclidean distance metric is not very robust to outliers and other distortions. In the proposed formulation, feature vectors from all frequency bands will have the same length and can be stored in a single database, and the feature vectors are designed to be robust to outliers and distortion.

Further, it is difficult to implement a real-time version of many methods proposed in the literature [9; 10; 11; 12]. To create feature vectors in the above techniques, signals are usually sampled at regular intervals which are closely offset in a technique commonly known as the sliding window method. This technique results in information being stored redundantly across a large number of feature vectors, which will result in unnecessarily large databases when applied to road networks. Some methods such as LCSS [13] are robust to outliers and can be indexed for fast retrieval purposes but still have the drawback of using the sliding window framework like all the above methods. In the sliding window method, the signal length that is used to generate a feature vector is fixed, regardless of the variation within the signal. This 'one-size-fits-all' approach is adequate for signals which exhibit significant variation over "small" time scales (e.g. music). However, for map-based localization, there may be sections of road which are very smooth and which have little variation in pitch information. Thus a fixed length segment of the signal might not create adequately unique feature vectors.

C. *Algorithm*

The above drawbacks reveal a need for a different approach towards generating feature vectors for pitch-based localization. This section introduces the 'Multi Scale Extrema Features', which attempt to overcome the above drawbacks. The individual steps involved in generating this feature vector are shown in Fig. 2.

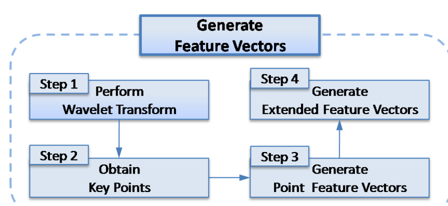

Fig. 2. The step by step process involved in obtaining the Scaled Shape Feature Vector.

1) Wavelet decomposition: To separate high-frequency noise from low-frequency features, wavelet decomposition is performed to partition the signal into its components corresponding to dyadic frequency bands. Next, feature vectors corresponding to each frequency band are computed. This method is used to restrict the effect of frequency-selective noise by limiting it to those feature vectors which have been extracted from the noise-affected frequency bands. The Wavelet transform is performed by using the so-called "Sombrero wavelet" whose Fourier transform is shown below.

$$\hat{\psi}(\omega) = \omega^2 e^{-\frac{w^2}{2}} \quad (1)$$

It has been shown that the wavelet transform [14] is equivalent to a multi scale differential operator.

$$Wf(u,s) = s^n \frac{d^n}{dx^n}(f * \theta_s)(u) \quad (2)$$

Where the wavelet $\psi(t) = (-1)^n \frac{d^n}{dx^n}(\theta(t))$ and $\theta(t)$ is typically chosen as the Gaussian function. In the present case n = 2, as we are utilizing the sombrero wavelet. Thus, the output signals, obtained from the wavelet transform, contain peaks corresponding to the high curvature points (second derivative) of different Gaussian-smoothed versions of the original signal. Step 1 in Fig. 3 presents an example of this wavelet transform of a signal.

2) Obtaining Key Points: Local maxima of the output from the wavelet transform are then selected as candidate "key points". These local maxima are calculated from the wavelet transform at each scale. This implies that, if a local maxima exists at time u_o and scale s_o , then:

$$\frac{\partial Wf(u,s)}{\partial u} = 0\bigg|_{u=u_o, s=s_o} \quad (3)$$

These key points are found from the finite-difference implementation of this equation. A heuristic measure for the susceptibility of a certain local maxima to noise is evaluated by measuring its distance to the neighboring peaks. An adaptive threshold for this heuristic measure is used to decide if a local maxima is to be designated as a key point.

Fig. 3 (step 2) shows that the key points of the wavelet transform at a particular scale represent high curvature points on the Gaussian smoothed version of the original signal at that particular scale. This entire process of taking the wavelet transform and finding the local maxima in the above manner is called 'Wavelet Modulus Maxima'[14]. By encoding the shape information at recognizable key points, this algorithm is able to achieve shift invariance. This procedure does away the need for encoding closely-offset overlapping frames, thus reducing the number of feature vectors required to encode a particular stretch of data.

An underlying assumption in this analysis is that the data is composed of different regions of constant slope (straight lines) and that the key points are the high curvature 'bridge' points between these straight lines. Road data tends to exhibit this effect, at least for the six thousand km measured by the authors.

3) Computing the point feature vector: Once the key points are obtained, the distance of a key point to its adjacent neighbors on the Gaussian smoothed version of the original signal is used to compute a point feature vector. By using the neighboring key points, the feature vector is able to expand to a scale suited to the underlying variation present in the signal. Thus, the signal length that is encoded is larger if the key points are far apart because of little variation in the data, and vice versa. This adaptive nature of the proposed feature vector enables it to overcome 'the one size fits all' restriction of current time series subsequence matching techniques.

For a one dimensional signal, the distance between two points in that signal is given by distance along the abscissa and the ordinate. Thus, four numerical quantities are required to describe the location of both the neighbors present on each side of a point. Let $a \; and \; c$ denote the distance of a key point along the abscissa to each of its neighbors and let b $and \; d$ be its distance along the ordinate to the same neighbors as shown in step3 of Fig. 3. Then the feature vector at that particular key point is given by:

$$[f(a,c)\; f(c,a)\; f(b,d)\; f(d,b)] \quad (4)$$
$$where \;\; f(x,y) = x/\sqrt{(x^2+y^2)}.$$

This encoding scheme allows the feature vector to be scale invariant with respect to the input data. Thus, a feature vector computed on a signal which is scaled by different amounts along the abscissa and the ordinate will be the same as that computed on the un-scaled signal. Only the relative distances between key points are used to compute the feature vector and this makes the resulting feature vectors bias invariant. Both bias and scale errors are commonly encountered when collecting pitch data and the proposed feature vector is designed to be immune to them.

4) Creating the extended feature vector: Finally, adjacent feature vectors are bundled together to create an extended feature vector in order to obtain an adequately unique representation of the shape around the key point. Choosing the length of an extended feature vector is a tradeoff between increasing the uniqueness of a feature and restricting the effect of an erroneous key point on the recognition of its neighborhood. Through implementation, it was found that at least three point features in each extended feature vector would be necessary for robust localization and an example extended feature vector of this nature is shown in step 4 of Fig. 3.

D. Feature Matching

Once the feature vectors are created they are used differently in the preprocessing phase and the online phase.

1) Preprocessing phase: In the pre-processing phase, the extended feature vectors are used to create a KD-tree in order to be able to perform an efficient search through the database of features. As the primary aim of this paper is to explore the efficacy of the feature vectors for localization, a generic tree was used for testing the vectors. A more detailed treatment of the various types of tree data structures that can be used for localization is presented in [15]. An interesting new data structure called vocabulary tree [8] has been reported to perform vision based localization very efficiently, and could easily be extended to the proposed method as well.

2) Online phase: In the online phase, the feature vectors are tested for a match within the database to determine their corresponding position estimate for the vehicle. Each query signal generates multiple feature vectors and each of these feature vectors is matched with the KD-tree database to determine their corresponding position estimate for the vehicle. Each position estimate was compiled into a histogram and the position with the highest value in the histogram is output as the best position estimate for a query signal. For applications in which local tracking follows the global localization scheme, the histogram can be used to output multiple position estimates which can be used to initiate a particle filter or multiple Kalman filters.

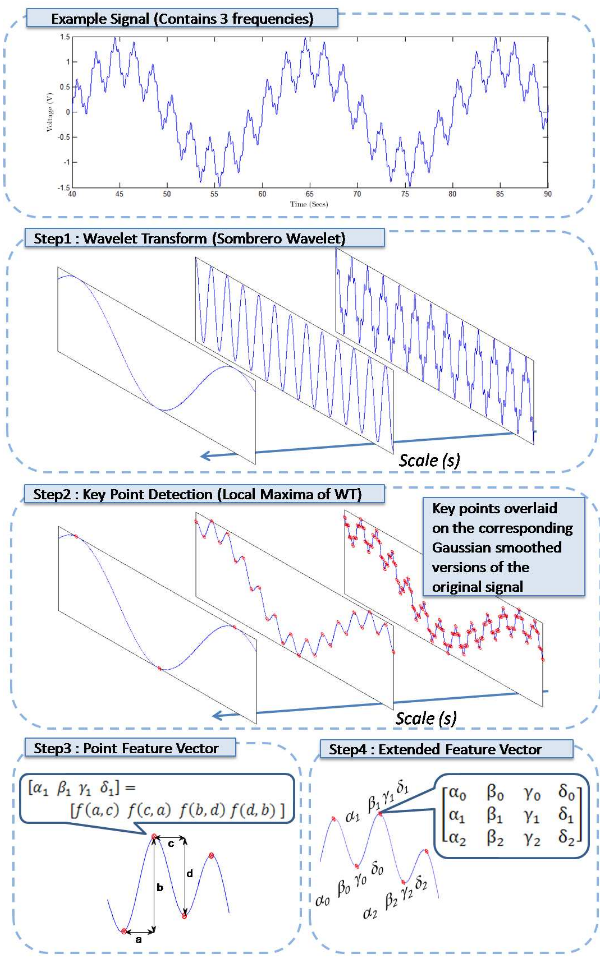

Fig. 3. The feature vector creation process by an example.

II. EXPERIMENTAL RESULTS

An experiment was performed on actual highways to evaluate the feasibility of this method for global localization. For the experiment, “map” data was collected once over 6000 km of roadway, and then “test” data was collected on a small portion of the same road way, across just 6 km. The full roadway, shown in Fig. 4, was used in the map building process while the second run was used in the testing procedure.

Fig.4. The roadway network that was used as a part of the experiments.

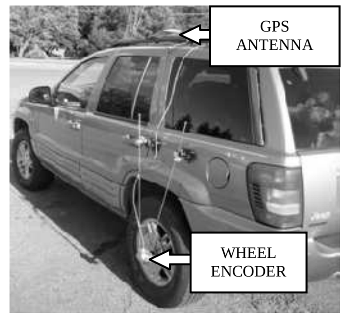

Fig.5. The figure shows the vehicle setup used in the experiments.

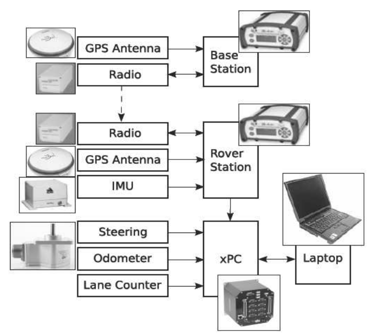

Fig.6. The figure shows the data acquisition system used in the experiments.

Thirty different query signals were extracted from the “test” data. To check the accuracy of localization, the ground truth for the 6 km stretch in the datasets was obtained from a DGPS system with a positional accuracy of 0.1 meters. It must be noted that while the mapping phase used the GPS information for representing the pitch information as a function of distance, the testing phase used the pitch information and the wheel encoder data to do the same. The localization estimate that is obtained from this representation of the pitch data is verified by using the GPS data that was collected as a part of the testing phase. A threshold of ten meters was used to determine if a certain match was accurate or not. Fig. 5 and Fig. 6 shows the experimental setup and Fig. 7 shows the accuracy curves that were obtained for the feature tree based method. The accuracy curves illustrate the localization accuracy that was obtained as a function of query signal length. It must be noted that for a query length of 800 meters, the correct position estimate was always amongst the first five position estimates that were obtained from the histogram of the position estimates. The mean and the standard deviation of the error in the location estimate, in the case of an accurate match for an 800 m query signal, were 1.96 meters and 0.68 meters respectively. It can be clearly seen that a threshold value of 10m is not very critical and slight changes to the threshold will not affect the accuracy curves in a significant manner. When tested on the 6000km database it was found that a single feature vector match took about 0.25 seconds. The entire matching process for a 800 meter signal took about 45 seconds on a 2.4Hz dual core computer when implemented by using a non-optimized MATLAB code.

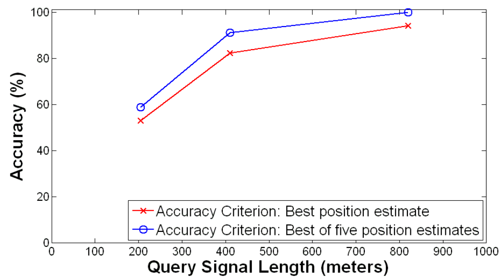

Fig. 7. Accuracy curves for localization under real conditions.

III. SIMULATION RESULTS

The objective of the simulation process is to test the ability of the designed feature vector to withstand various types of sensor noise typical of vehicle sensors and road measurement. Each type of sensor noise is represented by a corresponding parameter in the sensor model. In this paper, the pitch and the encoder sensors are modeled with the sensor models taken from [16] and [6], which are given by Eqns (5-6). It must be noted that the bias term for the pitch sensor error in [16] is modeled as a slow varying bias and is approximated as a constant for the purposes of this simulation.

TABLE I: Nomenclature for equations (5) and (6)

Symbol	Quantity
$Pitch_t$,$Pitch_m$	True pitch ,Measured pitch
B, S_f	Constant Bias error ,Constant Scale Factor error
v_{w1}	Zero mean band limited pitch noise
v_{w2}	Zero mean band limited encoder noise
$Encoder_t$,$Encoder_m$	True encoder value ,Measured Encoder value

$$Pitch_m = (1 + S_f)Pitch_t + B + v_{w1} \tag{5}$$

$$Encoder_m = Encoder_t + v_{w2} \tag{6}$$

These sensor models contain a total of four different error parameters (B, S_f, v_{w1}, v_{w2}), each of which represents particular types of noise. The error sources that are modeled by these parameters can come from both the sensor and the data collection process. For example, B includes the bias error in the pitch sensor and any inclination angle error in mounting the sensor to the vehicle. Similarly, v_{w1} and v_{w2} represents the zero-mean band limited white noise from the pitch and encoder sensors and also the noise from the vehicle chassis vibration and measurement errors due to slight differences in the lateral lane position of a vehicle during data collection. The bias and scale factor errors are expected to be mainly from the pitch sensor and the contributions from the data collection process are expected to be small if the sensors are properly mounted and calibrated. The standard deviation of the band limited noise for pitch data is again expected to be mainly from the sensor and the contribution from the data collection process was estimated to be 0.057 degrees (std dev) for the pitch data from experiments.

For this simulation, pitch data obtained from an integrated GPS-IMU system was used to generate feature vectors that were stored in a feature tree. Fig. 4 shows the 6000 km of roadway that was used in creation of the feature tree. Small portions of the original signal were taken and corrupted with each of the different noise types (B, S_f, v_{w1}, v_{w2}) up to varying degrees to create a "query signal". The feature vectors obtained from this query signal were matched with the original database to estimate the position in the database from which this query signal was extracted. The correctness of this position estimate was decided on the basis of a threshold distance (10 m) from the true point of extraction of the signal, e.g. any final estimate within this threshold is considered correct (local tracking algorithms can "lock" easily within this range).

For each of the four parameters, the simulation was performed by varying the parameter of interest while keeping others constant at their expected value for low cost sensors. The query signal was extracted from sixty different points of the original signal. Each query signal was corrupted and tested fifteen times in order to obtain a statistical estimate of algorithm performance that accounts for the random nature of the errors introduced. The tests were performed for query signals of lengths 205, 410 and 820 meters.

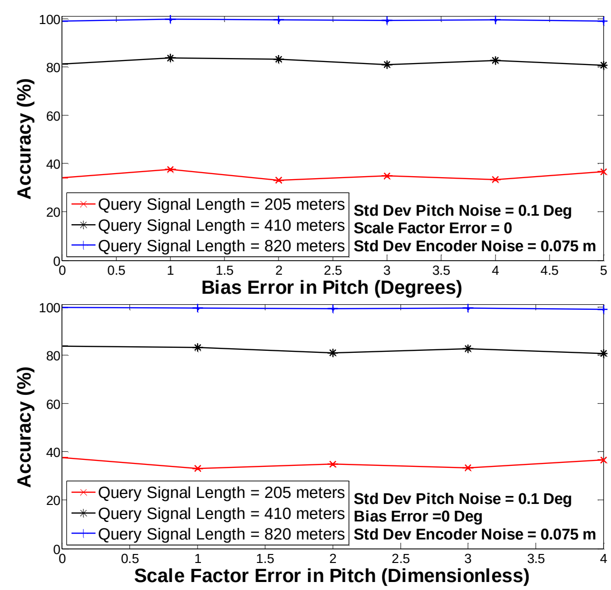

Fig. 8. The plots show that the localization accuracy of the feature based method is immune to the effect of bias noise and scale factor error in the pitch sensor.

Fig. 8 shows that the estimation process is unaffected by bias (B) and scale errors (S_f), a result that was expected as the feature vector was designed for scale and bias invariance.

Fig. 9 (top) shows that the estimation procedure was largely invariant to distance measurement (encoder) noise (v_{w1}) that one would encounter at highways speeds (60mph) which was estimated to be 0.076m (std dev) for each encoder tick at 100 Hz [6]. The addition of band-limited random noise in the pitch sensor (v_{w2}) was also investigated (Fig. 9, bottom). In [16], it was found that low-cost sensors used for pitch measurement had a standard deviation of v_{w2} of 0.1 degrees, so variations in pitch noise around this deviation were considered. This noise type appears to have a significant effect on the accuracy of the result.

Both Fig. 8 and Fig. 9 consider localization performance over several different query lengths. The results show that the performance of a more accurate pitch sensor can be achieved by a low-cost pitch sensor if one simply collects data over a longer period of time to obtain a longer query signal.

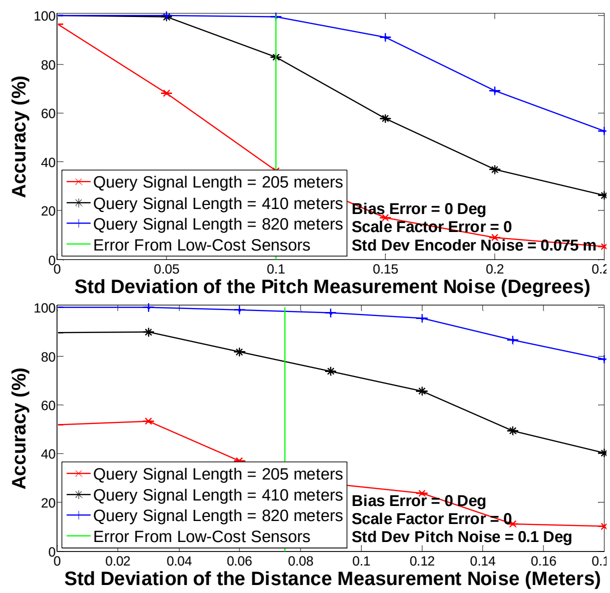

Fig. 9. The plots examine the effects of band-limited white noise in the encoder and pitch measurements on localization accuracy.

IV. Conclusions and Future Work

Firstly, the paper demonstrates the possibility of using pitch data for *global localization* in large roadway networks. By generating feature vectors for one dimensional pitch data, localization has been effectively performed for a road way network that is an order of magnitude larger than what has been previously demonstrated [3] [15]. Table II provides a comparison of the proposed method with localization methods using other sensors. While pitch information has low data density and hence can be utilized for localization over a large roadway network, the tradeoff is that a vehicle needs to travel a longer distance before localization is achieved.

This work also enables the use of other inertial measurements from vehicles, such as roll and yaw data, for localization. Future work could be directed towards implementing a feature vector that combines multiple sources of inertial data, thus reducing the query length because of the higher data density.

TABLE II: Comparison of different sensor modalities for localization

Sensor Citation	Map Size	Number of Features/Km	Travel distance for query signal
Vision [3]	20Km	5 x10^6/Km[a]	<1 m[a]
LIDAR [14]	165Km	9 x 10^3/Km[a]	~20-60 m[a]
Pitch [-]	6000Km	3 x 10^1/Km	~400 m

[a] Estimated from publication.

The paper also introduces 'Multi Scale Extrema Features' which are designed to overcome the expected drawbacks of using current time series subsequence matching techniques for inertial data. These features are robust to sensor noise and future work could involve demonstrating their capabilities by performing localization with low-cost inertial sensors.

Overall, this paper presents a promising new technique to perform global localization in order to compensate and/or replace GPS position estimates on roadway networks.

V. References

[1] Kang Li, Han-Shue Tan, and J. Karl Hedrick., "Map-Aided GPS/INS Localization Using a Low-Order Constrained Unscented Kalman Filter." Shanghai : IEEE Conference on Decision and Control, 2009.
[2] Bosse, M and Zlot, R., "Keypoint design and evaluation for place recognition in 2D lidar maps." Zurich, Switzerland : s.n., 2008.
[3] Schindler, G, Brown, M and Szeliski, R., "City-scale location recognition." Anchorage, Alaska : IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007.
[4] Dellaert, Frank, et al., "Monte Carlo Localization for Mobile Robots." Detroit, Michigan : IEEE International Conference on Robotics and Automation, 1999.
[5] Dean, Adam J, Vemulapalli, Pramod K and Brennan, Sean N., "Terrain-Based Road Vehicle Localization Using Particle Filters." Ann Arbor, Michigan : ASME Dynamic Systems and Control Conference, 2008.
[6] Dean, Adam J., *Terrain-based Road Vehicle Localization using Attitude Measurements.* State College, Pennsylvania : The Pennsylvania State University, 2008.
[7] Luke Ledwich, Stefan Williams., "Reduced SIFT Features For Image Retrieval and Indoor Localisation." Canberra, Australia : Australian Conference on Robotics and Automation, 2004.
[8] Stewénius, D. Nistér and H., "Scalable Recognition with a Vocabulary Tree." New York City : Computer Vision and Pattern Recognition , 2006.
[9] Faloutsos C, Ranganathan M, Manolopoulos., "Fast subsequence matching in time-series databases." Minneapolis : ACM SIGMOD conference, 1994.
[10] Ye, L and Keogh, E., "Time series shapelets: a novel technique that allow accurate, interpretable and fast classification." s.l. : Springer, 2010, Issue Data Mining and Knowledge Discovery .
[11] Keogh, Eamonn J. and Ratanamahatana, Chotirat., "Exact indexing of dynamic time warping." 2005, Issue Knowl. Inf. Syst, Vol. 7.
[12] Kin-pong, Chan and Ada Wai-chee, Fu., "Efficient Time Series Matching by Wavelets." Washington, DC : s.n., 1999, Issue Proceedings of the 15th International Conference on Data Engineering.
[13] Vlachos, M., Kollios, G. and Gunopulos., D., "Discovering similar multidimensional trajectories." s.l. : International Conference on Data Engineering, 2002.
[14] Mallat, S., *A Wavelet Tour of Signal Processing.* New York : Academic, 1999.
[15] Zlot, Robert and Michael, Bosse., "Place recognition using keypoint similarities in 2D lidar maps." Athens, Greece : International Symposium on Experimental Robotics, 2008.
[16] Elkaim, G, Lizarraga, M and Pedersen, L., "Comparison of Low-Cost GPS/INS Sensors for Autonomous Vehicle Applications." Monterey, Canada : ION/IEEE Position, Location, and Navigation Symposium, 2008.