# PATTERN MATCHING OF IN-VEHICLE ACCELERATION TIME SERIES DATA

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

Sean.N.Brennan Department of Mechanical Engineering The Pennsylvania State University

University Park, PA 16802 snb10@engr.psu.edu

# ABSTRACT

This paper presents a novel approach to find patterns in vehicle x-v-z acceleration data for use in prognostics and diagnostics. In this problem, vehicles are assumed to travel on the same routes and often times as a part of convoys but their GPS and other position information has been removed for privacy reasons. The goal of the pattern matching scheme is to identify the route or convoy associations within vehicles by using the acceleration data collected onboard these vehicles. A crucial step in solving this problem is to choose the right feature vector, as raw matching of acceleration signals is inappropriate due to different velocities, driving behaviors, vehicle loading, etc. In this paper, we demonstrate the feasibility of using 'Multi-Scale Extrema Features' for this application. The paper also addresses implementation details to enhance performance for in-vehicle acceleration data, corrupted by different sources of noise. A novel 'Multi-Scale Encoding' method is also proposed for the above feature vector and it leads to a significant improvement in the performance over traditional pattern matching methods. While the main focus of the paper is towards identifying feature vectors that effectively describe in-vehicle acceleration data, the feature vector could potentially be used with acceleration data obtained from other applications.

# **1 INTRODUCTION**

Previous work that identified patterns in acceleration data was aimed at identifying the nature of terrain that the vehicle was travelling on [1] or identifying the driver behavior [2]. Recently, acceleration data is also being used to identify the Gregory.S.Ledva Department of Mechanical Engineering The Pennsylvania State University University Park, PA 16802 gsl5013@psu.edu

Karl.M.Reichard Applied Research Laboratory The Pennsylvania State University University Park, PA 16802 kmr5@arl.psu.edu

activity of a person [3] as many smartphones come with accelerometers built into them. In this work we aim to extract location information by correlating patterns from acceleration data across multiple vehicles. This problem was encountered as a part of a broader research project where a large set of vehicle data (service records, acceleration data, etc) were collected over many years and on a number of different vehicles. While the acceleration data from these vehicles is known, the GPS data is unavailable due to privacy reasons. For purposes of prognostics and diagnostics, one of the tasks of the original project was to identify patterns in the data collected from fleet vehicles that were operated as a part of convoys or which were operated in the same routes. As the GPS information was unavailable, it was deemed necessary to explore the possibility of being able to cluster vehicles into common locations on the basis of measured acceleration data. If this grouping can be performed, then a possible application could be to pair this new information with the service records and other information from the vehicles to identify relationships between driving behavior patterns, positions, and repair histories.

Before solving the problem of clustering the vehicles, one must check feasibility of a signal-feature solution approach through a preliminary test where true positions are measured during the test. A preliminary test in this case would be to check whether the acceleration signatures of two vehicles that have travelled on the same road can be matched in the presence of vehicles. Also, the preliminary test data can be used to determine the most effective variant of a feature vector that can be used to solve the clustering problem. The main objective of this paper is to perform this preliminary test and identify feature vector formulations that will be effective in matching acceleration data collected from different vehicles.

The general overview of the preliminary test is as follows. Two sets of vehicle data including both acceleration and GPS (for ground truth) are collected on the same set of roads with two different vehicles. One set of acceleration data is used to create a database while portions of different length are extracted from the second dataset and are used to obtain a match to the first acceleration profile. The GPS locations from both the datasets are used to measure the accuracy of the matching process.

In its essence, the preliminary test consists of matching a signal with a large database of signals to find the most similar matches. This problem is often referred to as the subsequence matching problem [4]. The preliminary test is also similar to the map-based Global Localization problem in robotics [5,6,7]. In the Global Localization problem, a map (or database) is given to a robot, and robot must establish its location within the map by collecting sensor data and matching it with the map. Thus, this paper can be seen not only as a method to perform the feasibility test for grouping vehicles based on acceleration, but also as a feasibility test for acceleration-based localization.

A variety of sensors have been used for Global Localization over the years. LIDAR [5] and Vision sensors [6] are the most commonly used in robotics to perform Global Localization. Both LIDAR and Vision systems provide high dimensional data and the nature of these sensors is substantially different from inertial sensors such as an accelerometer and so the methods applicable cannot be directly utilized for pattern matching with acceleration data. In recent research, Vemulapalli et al [8] have reported that global localization can be performed using pitch data.

Global Localization with pitch data [8,9] has many similarities with respect to the 'preliminary test' problem using accelerations, but there are a number of key challenges that are specific to acceleration data. Pitch is generally easier to use because the pitch plotted against odometry does not change significantly with speed. The acceleration data, however, can undergo substantial distortion based on external conditions such as traffic. Moreover, the bias and scale factor variations in the pitch data are generally smaller than that for acceleration data.

While the paper utilizes the 'Multi-Scale Extrema Feature' vector framework developed for the pitch-based localization method, it evaluates different variants of the above feature vector and provides insights into the specific requirements and possibilities for acceleration data. This paper also proposes a novel 'Multi-Scale Encoding' method that enhances the performance of the feature matching algorithm. While the current work presents the results in the context of in-vehicle acceleration data matching, this feature vector could potentially be used for other acceleration matching applications.

Section 2 explains the challenges in matching acceleration

data from two different vehicles and presents a literature survey of the current subsequence matching techniques and their abilities to handle the above challenges. Section 3 presents the 'Multi-Scale Extrema' (MSE) features and the novel encoding method that has been utilized to solve this problem. Section 4 describes the experimental setup used to collect the data required to test the algorithm. Section 5 presents the results obtained from applying different variants of the MSE features to acceleration data. Conclusions then summarize the main results of this work.

# 2 BACKGROUND AND LITERATURE SURVEY

Before setting out to perform the 'preliminary test', one can visually verify whether acceleration data collected from two different vehicles on the same route have similar characteristics. Fig 1 shows the acceleration data collected on a vehicle that has travelled on a certain public route and within normal traffic patterns. One can clearly see the effect of the road layout on the acceleration data, wherein the turns of the route correlate to specific acceleration features. This implies that one can predict the acceleration of a vehicle, to a certain extent, based on the route that the vehicle is travelling on. Or conversely, one can use acceleration features to discern route location. The driver behavior, such as the speed at which one is travelling, external conditions, such as the traffic on the road, and the vehicle dynamics will also affect the acceleration data that is collected on a vehicle. This is in contrast to pitch based localization which is largely immune to these variations.



FIGURE 1. THE EFFECT OF THE ROUTE ON IN-VEHICLE ACCELERATION DATA

The key to acceleration-based pattern matching is the ability to extract the common features that can be matched irrespective of the nature of distortions that will be experienced due to human behavior and external conditions. Fig 2 shows the acceleration data collected on the same route as shown in Fig 1 on two different runs by different drivers. While one can definitely notice similarities between the data, the nature of distortions that one can observe is substantial: the five distortions that one can easily notice are shown in Fig 2 include

temporal distortion, outliers, bias distortion, scaling distortion, and random noise.

Given the nature of distortions in this case, a robust subsequence matching technique must be deployed. A number of different subsequence matching techniques have been proposed in the literature. Unfortunately, the similarity distance metrics that are used by these techniques have certain drawbacks that preclude them from being effective for acceleration data in particular. For example, the Euclidean distance has been reported as being brittle [10] to temporal distortions. Dynamic Time Warping (DTW) [10,11] has been introduced as a generalized form of Euclidean distance as it is robust to temporal distortion but it fails in the presence of outliers. This has led to 'Edit Distance' methods such as Edit Distance on Real Sequence (EDR) [11] and Longest Common Subsequence (LCSS) [12], which have drawn inspiration from methods used for matching strings in which dissimilar portions between the two strings are ignored. It has been reported in the literature that the edit distance methods are themselves sensitive to amplitude shifting and scaling [13]. The above methods [10,11,12] are also computationally burdensome because of the long length of the query signal in this application. Researchers have recently used local pattern based techniques [14,15], but most of these methods have relied on a sliding window approach and perform an exhaustive search across all window sizes and are thus computationally expensive.



FIGURE 2. THE DISTORTIONS THAT ARE EXHIBITED BY ACCELERATION DATA COLLECTED ON DIFFERENT RUNS.

The recently proposed MSE Features for pitch-based localization [8] have been designed to perform under a range of distortions without using a sliding window approach. The following section presents a brief overview of the subsequence matching algorithm and MSE Features. The section also presents a novel 'Multi-Scale Encoding' method which is a generalization of the sequential encoding method used in

previous work [8] and leads to better performance results.

#### **3 MULTI-SCALE EXTREMA FEATURES**

#### 3.1 Algorithm Overview

For the preliminary test, discussed in the first section, a query signal must be compared to a database of signals in order to obtain the locations of maximum agreement. Implementation of this algorithm can be divided into two phases: Preprocessing Phase and Testing Phase as shown in Fig 3. In the preprocessing phase, the signal-database is processed to obtain feature vectors, which are stored in an index-database (KDtree) along with the position from which each feature is extracted. In the testing phase, each query signal generates multiple feature vectors. Each of those features is matched to its closest neighbor, in terms of Euclidean distance, in the signaldatabase with the aid of the KD-tree. Each feature match yields an estimate for the position from which the feature is extracted. All such position estimates are compiled into a histogram and the position with the highest number of votes in the histogram is output as the best position estimate for the query signal.







FIGURE 4. THE STEP BY STEP PROCESS INVOLVED IN OBTAINING THE MULTI-SCALE EXTREMA FEATURES.

#### 3.2 Multi-Scale Extrema Features

This sub-section introduces the 'Multi-Scale Extrema Feature', which presents a very efficient wavelet scheme to generate feature vectors that capture the behavior of the signals over different resolutions. The individual steps involved in generating this feature vector are shown in Fig 4 and the corresponding descriptions are given below.

1) Wavelet decomposition: To separate high-frequency features from low-frequency features, wavelet decomposition is performed to partition the signal into its components corresponding to dyadic frequency bands. The wavelet transform is performed by using the so-called "Derivative of Gaussian (DoG) wavelet" whose Fourier transform is shown below.

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

Where  $\omega$  is a variable that denotes frequency.

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

$$\frac{\partial Wf(t,s)}{\partial t} = 0 \Big|_{t=t_o,s=s_o} \tag{2}$$

Where Wf(t, s) denotes the wavelet transformed version of the given signal and is a function of time 't' and wavelet scale 's'. Fig 5 (step 2) shows that the key points of the wavelet transform at different scales. This entire process of taking the wavelet transform and finding the local maxima in the above manner is called 'Wavelet Modulus Maxima' [16]. By encoding the shape information at recognizable key points, this algorithm is able to achieve shift invariance. This procedure does away the need for a sliding window approach, thus reducing the number of feature vectors required to encode a particular stretch of data.

Invariance	Feature Vector Encoding
Amplitude scale Amplitude Bias Time Scale	$\left[\frac{a}{\sqrt{a^2+c^2}}  \frac{c}{\sqrt{a^2+c^2}}  \frac{b}{\sqrt{b^2+d^2}}  \frac{d}{\sqrt{b^2+d^2}}\right]$
No Amplitude	[0 0 <i>b d</i> ]
Amplitude Bias	[ <i>a c b d</i> ]
Amplitude Bias Time Scale	$\begin{bmatrix} a & c & \frac{b}{\sqrt{b^2 + d^2}} & \frac{d}{\sqrt{b^2 + d^2}} \end{bmatrix}$

3) Computing the point feature vector: Once the key points are obtained, the distance of a key point to its adjacent neighbors 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 the sliding window technique.

Step3 of Fig 5 shows how the Point Feature Vectors (PFVs) are generated from distances between adjacent key points. Depending on the nature of invariance that one would like to incorporate, a feature vector could be encoded in a variety of ways as shown in Table 1. The parameters a, b, c, d shown in Table 1 are the relative distances of an extrema to its neighbors as shown in Step3 of Fig 5. The different PFVs in Table 1 are obtained by manipulating the distance parameters in order to remove certain types of information (Ex: Scale Information) in order to incorporate a certain invariance (Ex: Scale Invariance) into the vector. These vectors can be formulated in a number of different ways as discussed in Perng et.al [18]. The function f shown in step 3 of Fig 5 denotes a generic function used to encode the relative distance information into a dimension of the point feature vector and it does not have to be identical for all the different dimensions.



FIGURE 5. THE FEATURE VECTOR CREATION PROCESS BY AN EXAMPLE.

Table 1 offers different combinations of uniqueness and robustness that one could impart to the encoded feature vectors.

While one could define uniqueness and robustness in precise mathematical terms, an intuitive explanation is provided here for the sake of understanding. Uniqueness pertains to the extent to which a given feature vector differs from all the other feature vectors created from a particular dataset. On the other hand, robustness can be understood as the ability of a feature vector to remain close to its original form in spite of it being created from a signal which has undergone a certain form of distortion. As information is removed from a feature vector it becomes less unique and more robust. For example, as a hypothetical scenario one could imagine an encoding scheme where all elements of the feature vector are zeros irrespective of the distances of an extrema to its neighbors and this will result in feature vectors that are the most robust and the least unique.

The first encoding scheme in Table 1 allows the feature vector to be scale invariant (both temporally and amplitude wise) with respect to the input data. This robustness to scaling is provided because the scale information is removed from the encoded feature vectors and this leads to lesser uniqueness of the resulting feature vector. On the other hand, the third feature vector in Table 1 retains all the scale information and will therefore lead to a feature vector with higher uniqueness and lower robustness to scale distortions compared to the previously mentioned feature vector.

As only the relative distances between key points are used to compute the feature vector, all the above feature vectors are bias invariant. It is important to note that "Time Bias" invariance is not mentioned in Table 1 as that corresponds to the basic subsequence matching problem and is present in each one of those cases. Given the uncertain nature of variations present in the acceleration data, an experimental study was performed to choose the best-performing point feature vector. It was found that the amplitude bias feature vector gave the best performance for acceleration data. This implies that in spite of large distortions that were observed in the acceleration data (Fig 2), the uniqueness that is imparted from the scale information outweighs the robustness advantages accrued from eliminating the scale information. The experimental results to this effect are presented in section 5.1.

4) Creating the extended feature vector: Finally, adjacent Point Feature Vectors (P.F.Vs) are bundled together to create an extended feature vector in order to obtain an sufficiently unique representation of the signal's shape around the key point. Increasing the dimensionality of the feature vector or increasing the number of point feature vectors that combine to form an extended feature vector seems to be an easy way to increase the uniqueness of a feature vector and therefore achieve a better matching result. It is important to realize that this might not always be the case. Increasing the dimensionality implies that a longer signal is encoded into a feature vector and so any outlier or other distortion related artifacts are likely to contribute to a large number of feature vectors. Therefore, the increased information content due to high dimensionality could lead to reduced robustness. Thus, 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 in order to increase its robustness. For query signals of shorter lengths, there might not be sufficient information in certain wavelet scales in order to encode a feature vector and this may again decrease the accuracy. It is also important to bear in mind that increasing the dimensionality of the feature vector will result in a much larger computational effort while searching the KDtree because of the curse of dimensionality [17]. The dimensionality has to be determined experimentally and section 5.1 shows that utilizing three point feature vectors to build an extended feature vector gives the best pattern matching result for the vehicle data collected thus far. An example extended feature vector created from three point feature vectors is shown in step 4 of Fig. 5.

The extended feature vectors obtained from encoding adjacent Point Feature Vectors (P.F.Vs), as described above, will be referred to as Sequentially Encoded Multi-Scale Extrema features or SEMSE features. The next subsection presents a technique which allows for the encoding of nonadjacent Point Feature Vectors (P.F.Vs) and the features generated from this process are called Multi-Scale Encoded Multi-Scale Extrema features or MEMSE features.

# 3.3 Multi-Scale Encoding

Multi-Scale Encoding is a technique to improve the matching accuracy by encoding more feature vectors for a given signal that in turn captures more information about the signal. In the Multi-Scale Encoding method, point feature vectors from different scales are combined together to form extended feature vectors. Fig 6 illustrates the encoding mechanism for SEMSE and MEMSE feature vectors. The figure shows the features vectors that are formed with a point feature vector (P.F.V)(shown in a red glow) in conjunction with other P.F.Vs (shown in a yellow glow) for both the sequential encoding and Multi-Scale Encoding methods. Multi-Scale Encoding allows encoding of feature vectors from even those wavelet scales where there may be insufficient number of extrema in a particular scale to form a sequentially encoded feature vector. This leads to improved performance for shorter query signals. In this particular implementation, two point feature vectors (P.F.Vs) are combined to form a MEMSE feature and the amplitude bias invariant feature vector encoding from Table 1 is used to generate the P.F.V. As a large number of combinations of point feature vectors across multiple scales are possible, it becomes necessary to limit the number of combinations by setting time and scale windows in which suitable combinations can be found. Choosing a larger window size will lead to the creation of a larger number of a features but this would also increase the computational effort required for the pattern matching task.

In this particular paper, each point feature vector (P.F.V) from a given scale was combined with point feature vectors from two subsequent scales. Within these scales, the original P.F.V was combined with other P.F.Vs which were within a

certain time threshold interval from the original P.F.V. This time threshold has to be adaptive, as each wavelet scale represents the signal over different time lengths. The threshold for each scale was chosen to be twice the compact support of the wavelet filter at that particular scale. This allows for an adaptive threshold that adjusts itself to an appropriate extent corresponding to the filter.



FIGURE 6. AN ILLUSTRATION OF THE SEQUENTIAL ENCODING METHOD AND THE MULTI-SCALE ENCODING METHODS.

It can be seen that the Multi-Scale Encoding method is a generalization of the sequential encoding method, where the P.F.V's from different wavelet scales and beyond adjacent neighbors are combined. It is also important to note that the extended feature vector will contain two additional dimensions which store information about the difference in the scales and the temporal distance between the two combined P.F.Vs.

# **4 EXPERIMENTAL SETUP**

Acceleration and GPS data were collected along six predetermined routes. These routes ranged from 15-45 minutes in duration and included diverse driving conditions such as winding roads, mountainous roads, highways, downtown driving, etc. The total distance for all the six routes combined was 135 kms and the routes are shown in Fig 7 and Fig 8. Each of the six routes was driven in two different manners to test two particular scenarios.

1) Convoy Scenario: In the convoy situations, three cars drove the route simultaneously, with the cars safely following directly behind each other. The data collected in convoy scenario are expected to have similar characteristics as all the vehicles were travelling at similar speeds, in similar traffic conditions. However, there will be some variation due to the

#### different drivers involved.

2) Single Vehicle Scenario: In the single car situations, one car drove the route independently with no driving restrictions other than local traffic laws. The route was repeatedly driven under different times of the day (different traffic conditions). This provided a less controlled test where data collected was unique to driving style and traffic patterns.

The equipment used to collect data included a GlobalSat BU-353 GPS antenna sampling at 10 Hz, a SparkFun, three-axis, ADXL335 accelerometer sampling at 9600 Hz, a battery pack, and a data-logging box. The data-logging box housed the accelerometer and stored GPS and acceleration readings. The data-logging box was positioned behind the passenger seat and was firmly fixed to floor of the vehicle. The magnetized GPS antenna was mounted in the rear-window area of the car to provide higher satellite visibility. Throughout the tests, the equipment was positioned in the same orientation for data consistency. Refer to Fig. 8 for images of the equipment setup.



FIGURE 7. THE ROUTES COVERED AS A PART OF THE DATA COLLECTION EFFORT



FIGURE 8. THE SENSORS AND DATA ACQUISITION SYSTEMS USED IN THE EXPERIMENTS.

The GPS and accelerometer data were collected separately in the data-logging unit. Post-processing was used to convert the data into MATLAB data files. To compensate for the different sampling rates, the GPS and acceleration data were resampled to 10 Hz for further processing.

# **5 EXPERIMENTAL RESULTS**

This section presents the experimental results obtained by using Multi-Scale Extrema (MSE) features on the data from the two scenarios mentioned in the previous section. The tests are conducted using the subsequence matching procedure described in section 3.1. The testing procedure consists of a preprocessing phase in which the data collected on one of the vehicles is used to build the KD-tree data structure. In the testing phase, acceleration data obtained from another vehicle is used to create feature vectors and these features are matched with the KD-tree data structure. The position estimates from the matches are compiled into a histogram and the match with the highest number of votes provides a position estimate for the current vehicle. The accuracy of this estimate is compared to the separately measured GPS information and in this implementation, a distance threshold of 300 meters, in the database containing over 135000 meters of data, is used to verify if a resulting location estimate is accurate. A relatively lax threshold distance was utilized as this would be sufficient for the prognostics and diagnostics application which is the eventual target for the preliminary acceleration pattern matching problem.

As described earlier, the acceleration data collected on a vehicle depends on the route, driver behavior and external traffic conditions. Given these variations, the convoy acceleration data matching problem is easier because all these variations are expected to be similar as the vehicles are travelling in a convoy formation. On the other hand, in the nonconvoy acceleration data, only the variations due to the roadways are expected to match while the variations due to driver actions and external conditions are expected to be different and therefore inhibit the matching process. Due to these differences in the data types, one can notice that in all the subsequent tests, the accuracy result for the convoy dataset is higher than the accuracy for the non-convoy dataset. Therefore, the two datasets are useful to understand the behavior of the algorithms under different levels of noise. Given the two datasets, the next subsection presents evidence to support the parameter choices that have been made in constructing the feature vector. The subsequent subsection delves into the experimental results of the acceleration matching problem using different variants of Multi-Scale Extrema Features.

# **5.1 Parameter Tuning**

**5.1.1) Selecting the Point Feature Vector:** The first design choice in constructing the feature vectors is to choose a point feature vector from among the different options presented in Table 1. An experimental test was performed using the SEMSE features to decide the appropriate point feature vector from among those listed in Table 1 and the results are presented in Fig 9 and Fig 10. The test followed the methodology

described in section 3.1 and the two datasets described in the previous section were utilized.

The results for the convoy dataset are shown in Fig 9 and the Amplitude Bias encoding from Table 1 results in the best performance. These results can be intuitively explained as the convoy data is expected to matchup very well as all the effects such as route layout, the driver behavior and external conditions are expected to be similar for all the vehicles. This implies that because of the low noise situation, one would not require a high degree of robustness from the feature vector. Therefore, the Amplitude Bias point feature vector which provides a unique but not very robust feature vector would be very suitable for this situation.



FIGURE 10. ACCURACY CURVES FOR LOCALIZATION IN THE NON-CONVOY DATASET USING DIFFERENT TYPES OF FEATURE VECTORS.

Fig 10 shows that the "Amp-Bias" point feature vector performs well even in the non-convoy situation, but it must be noted that the "Amp bias, time scale" feature vector also performs well. It is quite likely that in the case when nonconvoy data is collected with large variations in speed, then "Amp bias, time scale" feature vector can outperform the "Amp bias" feature vector in the non-convoy scenario.

The results in Fig 9 and Fig 10 show that scale information in the feature vector makes an overall positive contribution to matching process because of the uniqueness it imparts in spite of the reduced robustness that might occur because of any scale factor variations between the matched signals.



FIGURE 11. EFFECT OF THE FEATURE VECTOR DIMENSIONALITY ON THE RETRIEVAL ACCURACY FOR THE CONVOY DATASET.



FIGURE 12. ACCURACY CURVES FOR LOCALIZATION IN THE NON-CONVOY DATASET USING DIFFERENT TYPES OF FEATURE VECTORS.

**5.1.2 Extended feature vector dimensionality:** The number of point feature vectors that are utilized to construct an extended feature vector is another important design choice that determines the dimensionality of the extended feature vector. Fig 11 and Fig 12 show the effects of the feature vector dimension on the retrieval result for the case of SEMSE features. It can be seen that, for both the datasets, the method in which the extended feature vector has three point feature vectors outperforms the other cases. The performance of the 1 point feature vector case can be attributed to the lack of adequate uniqueness in the feature vector. On the other hand, the performance of the 5 point feature vectors case can be attributed to the decrease in robustness as an erroneous artifact such as an outlier is encoded into a larger number of feature vectors.



FIGURE 13. ACCURACY CURVES FOR LOCALIZATION IN THE CONVOY DATASET USING DIFFERENT TYPES OF FEATURE VECTORS.



FIGURE 14. ACCURACY CURVES FOR LOCALIZATION IN THE NON-CONVOY DATASET USING DIFFERENT TYPES OF FEATURE VECTORS.

### **5.2 Experiments**

The pattern matching experiments are divided into two cases. In the single axis acceleration matching case, only the forward acceleration data is utilized in the matching process. On the other hand, the three axis acceleration matching case utilizes data from all X-Y-Z accelerations of a vehicle.

**5.2.1 Single Axis Acceleration Matching:** The pattern matching results comparing the SEMSE, MEMSE feature based matching and the traditional Euclidean distance method are shown in Fig 11 and Fig 12. In order to simplify the explanation, the results of SEMSE method are first compared with the Euclidean distance method and then a comparison between the MEMSE and SEMSE features is delineated.

1) SEMSE features vs. Euclidean distance method: The aim of this analysis is to compare the results of a particular implementation of the MSE feature (Sequentially encoded) with the Euclidean distance method by evaluating them on the same dataset. The sequentially encoded feature has been described in section 3.2. In this particular implementation, the amplitude bias invariant feature vector from Table 1 was chosen on the basis of the analysis performed in section 5.1.1. A total of three point feature vectors were used in each extended feature vector as this gave the best performance as shown in section 5.1.2. The dataset consists of acceleration data measured along a single axis and the results corresponding to convoy and non-convoy datasets are shown in Fig 11 and Fig 12 respectively.

The low noise level in the query data of the convoy dataset, results in the excellent performance of both the Euclidean and the Sequentially Encoded MSE (SEMSE) feature vector. However, one can notice that the SEMSE feature outperforms the Euclidean method at longer query lengths and this can be attributed to the non-robust nature of the Euclidean distance metric. It must also be noted that the Euclidean distance method performs better than the SEMSE feature vector for shorter query lengths as there may not be adequate number of extrema in shorter query signals in order to create unique feature vectors. In the case of the non-convoy dataset, the SEMSE feature outperforms the Euclidean distance based method because of its capacity to withstand complex deformations in a signal. The performance difference is stark especially with large query lengths, because of the ability of the MSE method to encode low frequency features which are very unique. While the DTW [18] based methods might result in better performance than the Euclidean data, the high computational demands of these methods makes them infeasible for the current application. It must be noted that MSE method not only results in better accuracy but is also computationally very efficient.

#### 2) MEMSE features vs. SEMSE features:

Fig 13 and Fig 14 also present the results of using different types of encoding techniques to build the extended feature

vector. The MEMSE feature vector clearly leads to better performance than the SEMSE feature vector, but the nature of the Multi-Scale Encoding technique leads to large number of feature vectors and this in turn leads to a slightly larger memory footprint and computational effort in this case.



FIGURE 16. ACCURACY CURVES USING 3-AXIS ACCELERATION DATA FOR THE NON-CONVOY DATASET.

**5.2.2 Three Axis Acceleration Matching:** A multiple KD-tree approach is utilized to incorporate acceleration data from different axis into the matching process. This method is similar to that presented in Section 3.1 except that three separate KD-trees are built to handle data along each axis. The position estimates from each KD-tree are assembled together into a single histogram that is used to decide the best position estimate. Fig 15 and Fig 16 show the results from this method for the convoy and the non-convoy datasets. One can clearly see that including new data sources into the matching process

improves the accuracy of all the methods and that the MEMSE feature vector gives the best performance in all the cases.

# **6 CONCLUSION AND FUTURE WORK**

Overall, the paper demonstrates the utility of Multi-Scale Extrema Features for encoding acceleration data. The paper also proposes the Multi-Scale Encoding method which leads to improvements in the performance under certain conditions, when compared to the sequentially encoded method. This analysis has shown that, given long query signals, the acceleration data from a vehicle travelling on a particular road can be matched in spite of differences in driver behavior and traffic conditions.

The performance advantages of using the feature vectors are clear, especially, in the case of longer query signals. The feature vectors that have been developed can not only be used for the originally mentioned clustering task but also be applied to other pattern recognition applications which rely on invehicle acceleration data.

An interesting direction of future work would be to extract extrema from Iterative Mode Functions (IMFs) obtained from Emperical Mode Decomposition (EMD) [20] because of their ability to handle nonlinear and non-stationary characteristics that was observed in acceleration data. Future work could also be directed towards minimizing driver variability effects by preprocessing or by utilizing encoder data in order to correct for rate of travel through feature sets. Additionally, because the application of these results considers reliability of the same vehicle model in operation, all testing was done using identical Chevy Malibus as the fleet vehicles and therefore the present study does not account for distortions in the acceleration data due to the different dynamics exhibited by different vehicle types. Work is ongoing to study the dynamic influence of vehicle-to-vehicle differences.

#### ACKNOWLEDGMENTS

We would like to thank Sanket Amin and Matthew A. Rigdon for their help in the data collection process.

# REFERENCES

- [1] D. Sadhukhan, C. Moore and E. Collins, "Terrain Estimation using Internal Sensors," *Robotics and Applications*, 2004.
- [2] A. Naito, C. Miyajima, T. Nishino, N. Kitaoka and K. Takeda, "Driver Evaluation Based on Classification of Rapid Decelerating Patterns," *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2009.
- [3] K. Laerhoven, E. Berlin and B. Schiele, "Enabling Efficient Time Series Analysis for Wearable Activity Data," in *Proceedings of* the 8th International Conference on Machine Learning and Applications, Miami, 2009.
- [4] C. Faloutsos, M. Ranganathan and Y. Manolopoulo, "Fast subsequence matching in time-series databases," in ACM SIGMOD international conference on Management of data, Minneapolis, 1994.

- [5] M. Bosse and R. Zlot, "Keypoint design and evaluation for place recognition in 2D lidar maps," in *Robotics and Autonomous Systems*, Amsterdam, The Netherlands, 2009.
- [6] G. Schindler, M. Brown and R. Szeliski, "City-scale location recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, 2007.
- [7] A. Dean, P. Vemulapalli and S. Brennan, "Highway Evaluation of Terrain-Aided Localization Using Particle Filters," in *Proceedings of the 2008 ASME Dynamic Systems and Control Conference*, Ann Arbor, MI, 2008.
- [8] P. Vemulapalli, A. Dean and S. Brennan, "Pitch based Vehicle Localization using Time Series Subsequence Matching with Multi-scale Extrema Features," in *American Control Conference*, San Franscisco, 2011.
- [9] S. Kadetotad, P. Vemulapalli and S. and Brennan, "Terrain-aided Localization using Feature-based Particle Filtering," in *Dynamic Systems and Control Conference*, Arlington, VA, 2011.
- [10] E. J. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, 2005.
- [11] D. J. Berndt and J. Clifford, "Finding Patterns in Time Series: A Dynamic Programming Approach," Advances in Knowledge Discovery and Data Mining, 1996.
- [12] L. Chen, M. Ozsu and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 28th international conference on Very Large Data Bases*, Hong Kong, 2005.
- [13] M. Vlachos, G. Kollios and D. Gunopulos., "Discovering similar multidimensional trajectories," in *Proceedings. 18th International Conference on Data Engineering*, San Jose, 2002.
- [14] Y. Chen, M. A. Nascimento, B. C. Ooi and A. K. H. Tung, "SpADe: On Shape-based Pattern Detection in Streaming Time Series," in *International Conference on Data Engineering*, Los Alamitos, 2007.
- [15] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allow accurate, interpretable and fast classification," in *Proceedings of the 15th ACM SIGKDD international conference* on Knowledge discovery and data mining, Paris, 2010.
- [16] T.-c. Fu, F.-l. Chung, R. Luk and C.-m. Ng, "Stock time series pattern matching: Template-based vs. rule-based approaches," in *Engineering Applications of Artificial Intelligence*, 2007.
- [17] S. Mallat, A Wavelet Tour of Signal Processing, New York : Academic, 1999.
- [18] C.-S. Perng, H. Wang, S. R. Zhan and D. S. Parker, "Landmarks: A New Model for Similarity-Based Pattern Querying in Time Series Databases," vol. 16th International Conference on Data Engineering (ICDE'00), no. Data Engineering, 2000.
- [19] R. E. Bellman, Dynamic Programming, 1957: Princeton University Press.
- [20] N. E. Huang et.al, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903-995, 1998.