Low-Cost Realtime Horizontal Curve Detection Using Inertial Sensors of a Smartphone

Shaohu Zhang¹, Myounggyu Won¹, and Sang H. Son²
¹WENS Lab, South Dakota State University, Brookings, SD, United States
²CPS Global Center, Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea
{shaohu.zhang,myounggyu.won}@sdstate.edu, son@dgist.ac.kr

Abstract—Fatal accidents occur frequently on low-volume rural roads, and the accident rates are up to 4 times higher at curves. It is thus of paramount importance to perform road inventory of rural roads to develop safety plans. However, most states in US face a challenge to maintain a database for low-volume rural roads due to limited funds for road inventory. In this paper, we propose to significantly reduce the cost for road inventory specifically focusing on horizontal curve detection by developing a mobile road inventory system based on off-the-shelf smartphones. The proposed system is capable of accurately detecting various kinds of horizontal curves by synthesizing heterogeneous smartphone sensor data to generate curve models by exploiting a machine learning technique. We implemented the system on iOS-based smartphones and tested with more than 400-miles of field data. We demonstrate that the proposed system achieves a median of 93.8% curve identification accuracy with a median of 5% false positive rates.

I. INTRODUCTION

Roadway curves are a potential safety hazard frequently causing roadway crashes. According to the Fatality Analysis Reporting System (FARS), 8,138 people were killed when negotiating roadway curves, which accounted for 21.4% of fatal crashes occurred along U.S. roadway in 2013 [1]. Identifying locations and geometric characteristics of the roadway curve is critical to the accident analysis and prevention. However, collecting curve information using traditional approaches is costly and time-consuming [2][3]. For example, it was reported that the average data collection cost per mile for photo/video log was $72 while $107 for satellite/aerial imagery [3]. Although some states maintain curve database including curve-related information (i.e., curve degree, curve direction and mileage) on U.S. and state highways, it is incomplete often not including curve information for rural roads [2]. There is high demand for road inventory of these rural roads.

Traditional approaches for road inventory rely on a survey vehicle equipped with sensors such as the scanner, GPS and the inertial system. Users are required to be familiarized with tedious device instructions and post-data-processing processes. Curve information may be more easily obtained using the geographic information system (GIS) map. However, as pointed out by Li et. al [2], low-quality GIS data with the transverse errors and low vertex resolution of the GIS roadway centerline results in curve identification errors. Figure 1 illustrates a typical curve with low-vertex resolution and irregular alignment lengths between two adjacent vertices that cause errors. High-resolution satellite imagery, e.g., 1m resolution, was used to extract curve information [4][5]. However, the accuracy greatly relies on image resolution and requires processing of a huge amount of images impeding the widespread use of satellite-based approaches [2].

Recent popularity of smartphones has prompted exponential growth of location-specific services and sensing applications [6][7]. In this paper, we propose a novel approach to roadway inventory based on embedded sensors of a smartphone to automatically identify and measure various kinds of curves. More precisely, we identify smartphone sensor data that are significantly correlated with curve detection, apply a machine learning technique to create model classifiers for various types of curves, and test the curve identification accuracy with more than 400-miles of field data.

The proposed system design, however, faces a number of challenges. First, the proposed system must be robust to measurement errors and outliers. Second, appropriate features for classification need to be selected to accurately identify curve types. Third, a smartphone has varying orientation within a vehicle. A mechanism needs to be developed to fix the coordinate system of a smartphone with that of a vehicle. In this paper, by addressing those challenges, we make the following contributions:

- We design and implement a cost-effective real-time mobile system that accurately and reliably detects various types of horizontal curves.
- We identify correlated features for curve detection and apply them to train our Support Vector Machine (SVM) models of curves.
- We perform large-scale field tests by collecting more than 400 miles of field data to train our model classifiers and to evaluate the performance of the proposed system.
II. SYSTEM DESIGN

A. Overview

The key idea is that the angular velocity around z-axis of the gyroscope (i.e., z-axis pointing toward the sky) is changed when a vehicle makes a curve; thus a curve can be identified by diagnosing variation in the angular velocity. Recently a similar approach has been proposed [8] in which thresholds of the angular velocity are defined to detect curves. We, however, note that the thresholds are dependent on vehicle speed. For example, Figure 2 shows that different vehicle speed results in varying patterns of the angular velocity, making it difficult to find appropriate thresholds taking into account the vehicle speed dynamics.

![Curve Detection](image)

Fig. 2. The angular speed on z-axis of a curve per varying vehicle speed

We address this challenge of the threshold-based approach by adopting a machine learning technique. In other words, we promote the approach to the curve detection problem, from the threshold-based static decision process to data-driven predictions that are known to effectively cope with dynamics, e.g., the vehicle speed dynamics. More specifically, the proposed system consists of three main components: Data Sampling, Noise Filtering, and SVM Engine as shown in Figure 3. The data sampling component collects a large amount of real-time sensor measurements and organizes the samples into a sliding window. In the noise filtering module, sensor data in a sliding window is filtered to remove noise and outliers. The filtered data is then provided as input to the SVM engine in which the data is compared with a classifier model for each curve type.

![Curve Detection System](image)

Fig. 3. An overview of the proposed curve detection system

B. Coordinate Alignment

The coordinate system of a phone comprises of three orthogonal axes $X_p$, $Y_p$, and $Z_p$ as illustrated in Figure 4.

![Coordinate Systems](image)

Fig. 4. The coordinate systems of a smartphone and a vehicle

The axes $X_p$, $Y_p$, and $Z_p$ point toward the left direction, the traveling direction, and the sky, respectively. When a vehicle makes a curve, it rotates around the z-axis, not making any rotation on x and y-axis. Thus we use the angular speed on $Z_p$ as a feature to detect curve types. A challenge is that a smartphone is usually carried in arbitrary orientation. Thus we need to align the coordinate system of a phone with that of a vehicle. In particular, we will align only the z-axis $Z_p$ of a smartphone with the z-axis $Z_v$ of a vehicle because we are concerned with only angular variations on the z-axis (Figure 4). To align the z-axis, we adopt a technique presented in [9]. The key idea is that since $Z_v$ is aligned with gravity the inclination angle between $Z_p$ and $Z_v$ can be estimated using the accelerometer.

More specifically, a gravity vector denoted by $g_p = [a_x, a_y, a_z]^T$ is derived from the accelerometer. Let $\hat{g}_p$ be a unit vector along $g_p$, and let $\hat{g}$ be a unit vector along $g$, $\hat{g} = [0, 0, 1]^T$. A quaternion $q$ that rotates $Z_p$ to $Z_v$ through the shortest arc can be calculated as follows.

$$
q = \cos \frac{1}{2} \theta + \hat{u} \sin \frac{1}{2} \theta
$$

$$
\hat{u} = \hat{g}_p \times \hat{g}
$$

$$
\hat{v} = \frac{1}{||u||} u
$$

$$
\cos \frac{1}{2} \theta = \sqrt{1 + v}
$$

$$
\sin \frac{1}{2} \theta = \frac{1 - v}{2}
$$

Using Hamilton product [10], the aligned output of the gyroscope $\omega_r = [\omega_x, \omega_y, \omega_z]$ can be calculated as,

$$
\omega_r = q \omega q^{-1}
$$

where $q = \cos \frac{1}{2} \theta - \hat{u} \sin \frac{1}{2} \theta$. We refer interested readers for more detail about the z-axis alignment to [9].

C. Measurement Noise Reduction

To achieve accurate curve detection, it is crucial to minimize measurement noise and outliers. To understand the nature of the measurement noise, we examined the distribution of noise. Figure 7 shows the histogram of the measurement noise, which is very close to the Gaussian distribution. The linearity of points in the QQ plot confirms that the measurement noise follows the Gaussian distribution. The Kalman filter is known to minimize the mean square error when the noise follows
Gaussian [11]. Thus we employ the filter to remove the noise. Figures 7 to 10 illustrate the noisy measurements and Kalman-filtered data of various curve types showing that the noise is effectively removed.

D. Feature Selection

In training our model classifiers in SVM, we aim to achieve higher detection accuracy by identifying additional features in addition to the angular velocity around z-axis. For this purpose, we adopt Spearman’s Rank-Order Correlation [12] to measure the strength of correlation between curve types and various kinds of sensor measurements such as vehicle speed, vehicle direction, and acceleration.

Our statistical analysis shows that the change of vehicle direction and acceleration on x-axis are highly correlated with curve types. Reasons for these results are attributed to the fact that when a vehicle makes a curve, the vehicle heading is changed, and the acceleration is also increased due to vehicle’s circular motion. In Section III-B, we will select an appropriate combination of the attributes experimentally.

III. Evaluation

In this section, we evaluate the performance of the proposed system. We first choose the optimal combination of features for training classifiers and select a sliding window size. The accuracy and false positive (FP) rates of curve identification are measured based on selected features and sliding window size.

A. Experiment Setup

<table>
<thead>
<tr>
<th>Type</th>
<th>Seg</th>
<th>Median</th>
<th>Mean</th>
<th>2.5%</th>
<th>25%</th>
<th>75%</th>
<th>97.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>860</td>
<td>-0.3239</td>
<td>-0.3234</td>
<td>-0.5551</td>
<td>-0.4186</td>
<td>-0.2170</td>
<td>-0.1123</td>
</tr>
<tr>
<td>CWC</td>
<td>4720</td>
<td>-0.0463</td>
<td>-0.0332</td>
<td>-0.0898</td>
<td>-0.0663</td>
<td>-0.0332</td>
<td>-0.0158</td>
</tr>
<tr>
<td>ST</td>
<td>10360</td>
<td>0.0300</td>
<td>0.0333</td>
<td>0.0121</td>
<td>0.0237</td>
<td>0.0398</td>
<td>0.0686</td>
</tr>
<tr>
<td>CCC</td>
<td>5060</td>
<td>0.0300</td>
<td>0.0333</td>
<td>0.0121</td>
<td>0.0237</td>
<td>0.0398</td>
<td>0.0686</td>
</tr>
<tr>
<td>LT</td>
<td>940</td>
<td>0.2933</td>
<td>0.2837</td>
<td>0.0626</td>
<td>0.2143</td>
<td>0.3650</td>
<td>0.4471</td>
</tr>
</tbody>
</table>

We collected more than 400 miles of field data on state and rural highways, and local roads. The collected data were used for training SVM models and for examining the performance of the proposed system. Table I summarizes the angular velocity around z-axis (the main feature) of the field data. Each ‘segment’ represents sensor measurements for 1 second. More specifically we used the C-support vector classification [13] for training SVM models, and we considered the 2.5th percentile as the lower boundary and the 97.5th percentile as the upper boundary for classifying curve types into 5 categories: Clockwise Curve (CWC), Counter Clockwise Curve (CCC), Straight Road (ST), Left Turn (LT), and Right Turn (RT). We selected 20 LT’s, 15 RT’s, 10 CCC’s, and 12 CWC’s to test the performance of the proposed system (Figure 11).

The proposed system was implemented as a mobile application on iOS 8 running on iPhone 4s or later. To offload the computation overhead for training our models, the SVM models were generated off-line and were stored on smartphones. Sensor data was continually collected on a smartphone and was compared with the model classifiers individually for each curve type. Once a curve was detected, its information such as location, length, types, etc., was stored in a file for later analysis.
B. Effect of Parameter Selection

![Fig. 12. Accuracy per feature combination for curves](image1)

To determine the optimal combination of features for training classifiers, we measure the detection accuracy with varying combinations of features. Specifically, we considered the features that are highly correlated with curve types: A (acceleration on X axis), Z (rotation rate around Z axis), and D (change in the vehicle heading angle). For this experiment, the window size was fixed to 2 seconds (Justification will be presented in Section III-C). Figures 12 and 13 show the identification accuracy with different combinations. We observed that all combinations had high detection accuracy. An interesting observation was that only when all features are used, i.e., ZAD, the highest accuracy was achieved, while a single feature used with feature Z did not yield much improvement.

C. Selection of Window Size

![Fig. 14. Effect of window size for curves](image2)

We determine the window size for this experiment. We measured the detection accuracy for curves and turns with varying window sizes. We note that the best window size depends on the curve length and vehicle speed. For example, turns are completed relatively quickly; thus large window size results in lower accuracy as illustrated in Figure 15. The results show that the window size can be dynamically selected and multiple window sizes may be used for more accurate curve detection. Due to space constraints, we leave this dynamic multi-dimensional window size selection mechanism as our future work. In this experiment, we found that a window size of 2 second gives overall good performance for both curves and turns (Figures 14 and 15).

D. Accuracy of Curve Detection

![Fig. 16. Accuracy for curve types](image3)

Having selected the window size and the feature combination, we now measure the accuracy of different types of curves. The box plots in Figures 16 and 17 show the results. The median accuracy for all types of curves were greater than 90%. The 5th percentile was 91.3%, 83.3%, 86.6%, and 80% for CWC, CCC, LT, and RT, respectively, showing reliable accuracy for nearly all sample curves.

E. False Positive Rates

![Fig. 18. False positive rates for curve types](image4)

![Fig. 19. False positive rates for turn types](image5)

When a certain type of curve is detected, it is possible that other types of curves may also be detected. We measure these false positive rates for each curve type. It was observed that the false positive rates were low for all types of curves (Figures 18 and 19). Specifically, the mean false positive rates were 1.7%, 2.8%, 8%, and 7.5% for CWC, CCC, LT, and RT, respectively. An interesting observation was that turns had higher false positive rates because turns are typically completed more quickly than curves.

IV. RELATED WORK

For the purpose of roadway maintenance and safety, many researchers focused on the extraction and identification of curve information. The most common methods employed involve the use of survey GPS data, satellite imagery, and geographic information systems (GIS) tools.

**Field Survey based Approach:** Some researchers have proposed GPS-based procedures to identify horizontal curves [14][15][16][17]. In this approach, a survey vehicle equipped with a GPS receiver and other sensors like scanners is used to obtain geographical coordinates of curves. Collected raw GPS data are post-processed to extract curve information. For example, Kim et al. used the terrestrial laser scanning technology to extract horizontal and vertical alignment of a curve [18]. However, these approaches not only incur high costs but require users to be familiarized with operating devices/equipments and post-processing raw data.
Image Processing based Approach: High-resolution satellite imagery has been used to extract roadway curve information [19][20][21][22]. Easa et al. designed a method using IKONOS 1m spatial resolution imagery to extract simple circular and reversed curves [4]. Dong et al. applied the Hough Transform algorithm to develop an approximate method for extracting spiral horizontal curves using high-resolution satellite imagery [5]. Although these approaches can retrieve geometric characteristics of some typical curves by using an approximate algorithm, the limitations are that the accuracy greatly relies on image resolution and that it requires processing of a huge amount of high-resolution images incurring high computational overhead.

GIS based Tool: There are three main GIS-based methods: Curve Calculator [23], Curvature Extension [24] and Curve Finder [25]. The Curve Calculator is developed by Environmental Systems Research Institute (ESRI). It allows users to manually define the beginning and ending of a curve and input any two of four curve characteristics (i.e., chord length, arc length, and radius) to generate one curve information at a time. The Curvature Extension is developed by the Florida Department of Transportation (FDOT). Similarly, users have to manually define the beginning and ending of a curve and input parameters of the curve to obtain a curve information at a time. The Curve Finder is a program developed by the New Hampshire Department of Transportation (NHDOT), which allows for an automated procedure that can be executed on a network of roadways. Curves are identified as the program moves through every series of points. Recently Li et al. demonstrated that the Curve Finder can be applied to low-volume rural roads from a selected roadway layer for classifying curves, computing curve geometries and creating a geographic information system for curve layers automatically [2]. Li et al. showed that GIS tools provide an inexpensive and efficient way to obtain curve information. However, the identification accuracy relies on high-quality GIS data. The inconsistency of roadway alignment in the GIS data requires users to train the data to manually determine the threshold.

V. CONCLUSION

We have presented the design, implementation and evaluation of the mobile system for low-cost and realtime road inventory specifically. We expect that with the proposed system, anyone can easily conduct road inventory on rural roads allowing for crowd sourcing for road inventory, i.e., numerous people conduct a survey thereby dramatically improving the reliability of the survey results. It also can be integrated with advanced driver assistance systems to control excessive adjustment to curves in realtime. Our future work includes incorporating other aspects of road inventory such as vertical curve detection.

VI. ACKNOWLEDGEMENT

This research was supported in part by the DGIST R&D Program of MSIP of Korea (CPS Global Center) and the Global Research Laboratory Program through NRF funded by MSIP of Korea (2013K1A1A2A02078326).

REFERENCES