

GEORGIA DOT RESEARCH PROJECT 15-05

FINAL REPORT

**CURVE IDENTIFICATION FOR
HIGH FRICTION SURFACE TREATMENT (HFST)
INSTALLATION RECOMMENDATION**



**OFFICE OF RESEARCH
15 KENNDY DRIVE
FOREST PARK, GA 30297-2534**

Contract Research

GDOT Research Project No. 15-05

Final Report

**CURVE IDENTIFICATION FOR HIGH FRICTION SURFACE
TREATMENT (HFST) INSTALLATION RECOMMENDATION**

By

Yichang (James) Tsai, Ph.D., P.E.

Chengbo Ai, Ph.D.

Yiching Wu, Research Engineer

Georgia Institute of Technology

Contract with

Georgia Department of Transportation

In cooperation with

U.S. Department of Transportation

Federal Highway Administration

September 2016

The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Georgia Department of Transportation or of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

1. Report No.: FHWA-GA-17-1505	2. Government Accession No.:	3. Recipient's Catalog No.:	
4. Title and Subtitle: Curve Identification for High Friction Surface Treatment (HFST) Installation Recommendation		5. Report Date:	
		6. Performing Organization Code:	
7. Author(s): Yichang (James) Tsai; Chengbo Ai; Yiching Wu		8. Performing Organ. Report No.: RP 15-05	
9. Performing Organization Name and Address: Georgia Institute of Technology 790 Atlantic Drive Atlanta, GA 30332-0355		10. Work Unit No.:	
		11. Contract or Grant No.: RP 15-05; PI# 0013631	
12. Sponsoring Agency Name and Address: Georgia Department of Transportation Office of Research 15 Kennedy Drive Forest Park, GA 30297-2534		13. Type of Report and Period Covered: Final;	
		14. Sponsoring Agency Code:	
15. Supplementary Notes: Prepared in cooperation with the U.S. Department of Transportation, Federal Highway Administration.			
16. Abstract: The objectives of this study are to develop and deploy a means for cost-effectively extracting curve information using the widely available GPS and GIS data to support high friction surface treatment (HFST) installation recommendations (i.e., start and end points of HFST). This is accomplished by developing a Smart Curve Information Extraction (Smart-CIE) tool for extracting curve information that is essential but often lacking for various safety analyses. The developed Smart-CIE tool consists of three core components, including a) an automated curve identification component that implements an algorithm previously developed by the authors to automatically extract curve information, including the curve location, radius, the point of curve (PC), and the point of tangent (PT) of every curve, b) a map-based quality assurance and quality control (QA/QC) component to incorporate engineers' knowledge of roadway geometry, and c) a map-based curve information reporting component. In addition, an extended component, HFST Start/End Points Identification, was developed in the SMART-CIE to determine the start and end points for HFST installation using the extracted PC and PT and the Texas Transportation Institute (TTI) HFST installation guideline. A case study, using GPS data collected on 31 miles of State Route 2 in Georgia, was conducted to demonstrate the superior capability of the Smart-CIE for extracting curve information in a reliable and efficient manner and the capability of accurately determining the HFST start and end points for a cost-effective HFST installation, which results in an estimated saving of \$600,000 on State Route 2.			
17. Key Words: High friction surface treatment; HFST; curve finder; GPS/GIS		18. Distribution Statement: No Restriction	
19. Security Classification (of this report): Unclassified	20. Security Classification (of this page): Unclassified	21. Number of Pages: 92	22. Price:

TABLE OF CONTENTS

LIST OF TABLES	iii
LIST OF FIGURES	v
ACKNOWLEDGMENTS	vii
EXECUTIVE SUMMARY	ix
1. INTRODUCTION	1
1.1 Research Background and Need	1
1.2 Research Objectives and Tasks	2
1.3 Organization of This Report.....	5
2. LITERATURE REVIEW OF CURVE IDENTIFICATION METHODS	7
2.1 Manual Curve Identification Methods	7
2.2 Automatic Curve Identification Methods.....	8
2.3 Implementation of Curve Identification Methods.....	10
2.4 Image-Based Method	14
2.5 Summary	14
3. SMART-CIE: AN SMART CURVE INFORMATION EXTRACTION TOOL	17
3.1 Overview of the Smart-CIE Tool.....	17
3.2 Automated Curve Identification Component	19
3.2.1. Algorithm Overview	20
3.2.2. Operation Flow	22
3.2.3. Validation of the Automated Curve Identification	26
3.3 Map-Based QA/QC Component	30
3.4 Map-Based Curve Information Reporting Component.....	33
3.5 Case Study on State Route 2	33

3.5.1.	Data Collection	34
3.5.2.	Data Processing Procedure	35
3.5.3.	Curve Inventory Results	36
3.6.	Summary	38
4.	SMART-CIE EXTENDED COMPONENT: HFST START/END POINTS IDENTIFICATION FOR HFST INSTALLATION RECOMMENDATION	39
4.1.	Identification of the Start/End Points of HFST	39
4.2.	HFST Start/End Points Identification Component.....	41
4.3.	Case Study on State Route 2	43
5.	CONCLUSIONS AND RECOMMENDATIONS	47
	REFERENCES	51
Appendix A	DETAILED CURVE FINDER ALGORITHM	A-1
Appendix B	KASA’S METHOD	B-1
Appendix C	CLOTHOID SPLINE METHOD.....	C-1
Appendix D	PROCEDURE FOR USING ARCOBJECTS ADD-IN APPLICATION	D-1

LIST OF TABLES

Table 2-1 Summary of automatic curve identification methods.....	10
Table 3-1 Confusion matrix for curve classification test.....	27
Table 4-1 Look up table for determining the HFST starting point before PC.....	40

LIST OF FIGURES

Figure 2-1 The interface for the Curve Extension by FDOT	11
Figure 2-2 The interface for the Curve Finder by NHDOT	12
Figure 2-3 The interface for the CurveFinder by WisDOT	13
Figure 3-1 Proposed operation flow of the Smart-CIE Tool	18
Figure 3-2 Illustration of horizontal curves delineated by PC and PT	21
Figure 3-3 Flowchart for the proposed automatic horizontal curve identification and measurement method	21
Figure 3-4 Import data	23
Figure 3-5 An example of input data	24
Figure 3-6 An example of the curves identified by the tool	25
Figure 3-7 Selected roadway in the field test.....	28
Figure 3-8 Curve radius measurement results of the field test	29
Figure 3-9 The operation of the Add functionality for the Smart-CIE tool.....	31
Figure 3-10 The operation of the Delete functionality for the Smart-CIE tool	32
Figure 3-11 The operation of the Merge functionality for the Smart-CIE tool	32
Figure 3-12 Overview of the case study roadway section on State Route 2.....	33
Figure 3-13 The data collection vehicle and the sample GPS trajectory data	34
Figure 3-14 The QA/QC interface for Smart-CIE in network-level analysis.....	35
Figure 3-15 Examples of undetected curves and incorrectly detected curves	37
Figure 4-1 Procedure of determining the HFST installation boundary in a curve.....	41
Figure 4-2 An example of determining the start/end points of HFST	42

Figure 4-3 Example of the HFST installation boundary on State Route 2 in PI#0009993	43
Figure 4-4 Examples of the identified curves with measurements on the State Route 2.....	44
Figure A-1 Pseudo code and illustration for iterative circular fitting algorithm	A-2
Figure A-2 Illustration of the resultant segments on a small road section.....	A-3
Figure A-3 Illustration of different center angles for curved and tangent segments	A-4
Figure A-4 Distribution of the center angles in the calibration dataset	A-5
Figure A-5 Illustration of the segment adjacency patterns for different curve types	A-7
Figure C-1 Illustration of the Clothoid spline with a back spiral	C-1
Figure D-1 Import data	D-1
Figure D-2 An example of selecting 'X Field' and 'Y Field' for CSV inputs.....	D-2
Figure D-3 The projection system automatically populates for shapefile input.....	D-2
Figure D-4 An example of selecting the output coordinate system.....	D-3
Figure D-5 An example of filling up Import Data forms after selecting the working.....	D-4
Figure D-6: An example of curves identified by the tool	D-5
Figure D-7 A visualization of the curves identified by the tool	D-5
Figure D-8 An example of selecting curves to locate HFST endpoints	D-6
Figure D-9 User inputs posted and advisory speed for each curve.....	D-6
Figure D-10 An example showing the populated HFST table.....	D-7
Figure D-11 A visualization of HFST endpoints.....	D-7
Figure D-12 Click 'Export' to save the results in local system	D-8

ACKNOWLEDGMENTS

We thank Mr. Andrew Heath, Mr. David Adams, Mr. Michael Turpeau, from the Office of Traffic Operations, for their strong support and heavy involvement. We would like to thank Mr. David Jared and Mr. Binh Bui from the Office of Research for their support. We would like also to thank Ms. Meg Perkle and Ms. Kathy Kaul for initiating this research project. We would like to thank the US DOT for its support on the research project “Remote Sensing and GIS-enabled Asset Management System (RS-GAMS),” which enabled us to use emerging 2D imaging, 3D light detection and ranging (LiDAR), and a global positioning system/inertial measurement unit (GPS/IMU) installed in the the Georgia Tech Sensing Vehicle (GTSV) to collect the detailed level of curve data used in this project. We would like to thank GDOT district engineers, Mr. Matt Needham, Mr. Chris York, and Mr. Jerry Thomason for their assistance with our data collection. We would also like to thank the members of the research team at the Georgia Institute of Technology (Georgia Tech), including Cibi Pranav and Geoffrey Price, for their diligent work on data collection, processing, and analysis in this research project.

EXECUTIVE SUMMARY

According to the Every Day Counts (EDC) initiative, more than 25 percent of fatal crashes in the United States occur on horizontal curves. Horizontal curve information, including curve location, radius, the point of a curve (PC), and the point of a tangent (PT), is essential but often lacking in support of curve safety analyses, such as determining high friction surface treatment (HFST) installation recommendations (i.e., the start and end points of HFST). The objective of this study is to develop and deploy a cost-effective means for extracting essential curve information, including curve location, radius, PC, and PT, by using widely available global positioning system (GPS) and geographic information system (GIS) data that support recommendations for HFST installation. This is accomplished by developing a smart curve information extraction (Smart-CIE) tool consisting of three core components: 1) an automated curve identification component, 2) a map-based quality assurance and quality control (QA/QC) component, and 3) a map-based curve information reporting component. This tool will facilitate the curve information extraction process. In addition, an extended component, HFST Start/End Points Identification, was developed to determine the start and end points for HFST installation using the extracted PC/PT and the HFST installation guidelines developed by the Texas Transportation Institute (TTI). A case study using the 31 miles of GPS data collected on State Route 2 in Georgia was conducted to demonstrate the capability of extraction curve information using the Smart-CIE and the potential savings of utilizing the HFST Start/End Points Identification to precisely determine the start and end points of HFST. The outcomes of this study are summarized as follows:

- A smart curve information extraction (Smart-CIE) tool has been developed to cost-effectively extract detailed curve information, including curve location, radius, PC, and PT,

using widely available GPS and GIS data. The Smart-CIE tool was developed in ArcGIS 10.3.1 as an add-in to streamline the curve identification process using the common ArcGIS platform. The Smart-CIE consists of three core components: a) an automated curve identification component that implements an algorithm previously developed by the authors to automatically extract curve information from the huge amount of available GPS and GIS data, b) a map-based QA/QC component to incorporate engineers' knowledge, and c) a map-based curve information reporting component.

- Validation using simulated and field data collected by the research team was conducted to critically evaluate the performance of the automated curve identification component of the Smart-CIE tool. The validation data contains 385 simulated curves with four curve types (simple, compound, reverse, and spiral) and 25 actual simple curves. The results demonstrated the accuracy of the algorithm (i.e., the correlation slope of 1.01 and the R^2 of 0.99 strongly suggest an accurate measurement) and showed significant improvement in the productivity of the curve information identification process.
- A case study on State Route 2 was conducted to demonstrate the Smart-CIE tool has the capability to conduct a network-level curve inventory efficiently. The tested section covers 31 miles of State Route 2 and consists of 92 horizontal curves. The processing time of the 31 survey miles (11,773 GPS points) takes less than 20 minutes (including the execution of the automated curve identification component at less than 30 seconds, the map-based QA/QC at less than 19 minutes, and the map-based result reporting at less than 5 seconds. More than 86% (80 out of 92 curves) of the curves were automatically extracted using the automated curve identification

component, and the remaining 12 curves were also effectively inventoried using the map-based QA/QC component (e.g., add, delete, and editing curve PC and PT). The results show that the Smart-CIE tool provides an accurate and efficient means for transportation agencies to extract the curve information at the network level, which is essential to support transportation agencies' curve safety improvement programs (e.g., SafetyAnalyst, model inventory of roadway elements (MIRE), etc.).

- An extended component, HFST Start/End Points Identification, was also developed in the Smart-CIE tool using the extracted PC/PT and the Texas Transportation Institute (TTI) HFST installation guidelines to reliably and accurately determine the start and end points for HFST installation. The results show it is promising for transportation agencies to use the HFST Start/End Points Identification component to consistently and accurately determine the starting and ending points of HFST installation, which will result in savings in HFST application.
 - A case study using the HFST sites in a GDOT project (Project Identification Number, PI#0009993) on State Route 2 was conducted to demonstrate the capability of the extended component of HFST Start/End Points Identification in accurately and effectively determining the start and end points of the HFST installation. Results show the developed component is very promising for saving HFST compared to the current manual procedure of determining the start and end points of the HFST installation. It has resulted in an estimated saving of \$600,000; it saves approximately 25,000 sq. yd. of HFST material and reduces operational costs that would have been required by the manual determination of the start and end points of HFST installation, even when the same TTI HFST installation guidelines are used.

The implementation of research outcomes and recommendations for future research are presented below:

- It is recommended that training be conducted for implementation of the Smart-CIE, including one training session for GDOT, one training session for counties and cities through the GDOT Local Technical Assistance Program (LTAP), and one nationwide training session through the Federal Highway Administration (FHWA) Every Day Counts (EDC) program. To better disseminate the results, the training sessions may be presented as traditional classroom training or webinar.
- It is recommended that GDOT conduct a further study to validate the curve information (e.g., curve radius, PC, and PT) extracted by different means, such as CARS (Curve Advisory Reporting Service) and Smart-CIE.
- It is recommended that GDOT implement a statewide curve inventory that uses the developed Smart-CIE to maximize the utilization of the GPS and GIS data already available to the Office of Transportation Data (OTD).
- It is also recommended that GDOT implement the extended component of HFST Start/End Points Identification HFST installation.
- It is recommended a smartphone app be developed to further improve the extended component of HFST Start/End Points Identification for locating the start and end points in the field.

1. INTRODUCTION

1.1 Research Background and Need

Horizontal curves remain one of the most over-represented locations for high-frequency, high-severity crashes. According to the Every Day Counts (EDC) initiative, more than 25% of fatal crashes in the United States occur on horizontal curves that account for 5% of roadways.

Transportation agencies make substantial investments and efforts to mitigate the crashes on curves by applying different safety improvement countermeasures; curve information, including location, radius, the point of curve (PC), and the point of tangent (PT), are essential for supporting the decisions involving these countermeasures. This is especially true for the high friction surface treatment (HFST), which uses high-quality, durable aggregates to immediately restore pavement friction at the locations where sufficient friction is most needed (e.g., on a sharp curve). Compared to the other countermeasures, such as rumble strips and signage, the HFST is a more expensive site-specific countermeasure. Thus, it is critical to install the HFST at the right places (i.e., the start and end points of HFST), and this decision relies on the PC/PT and speed. The HFST is typically extended outside PC and PT for a certain distance based on the speed. However, such essential curve information, including PC/PT, is usually lacking, although it plays an indispensable role in supporting the decisions on where and how to improve horizontal curve safety.

Traditionally, transportation agencies use a plan-reading method and a chord-offset method to determine a specific roadway curve radius in the field. Due to the nature of these methods, the process can be time-consuming and labor-intensive for field engineers. Unfortunately, it becomes increasingly infeasible for transportation agencies that maintain large road networks

and numerous horizontal curves. Thus, in recent years, transportation agencies have been seeking a cost-effective method for extracting curve information by leveraging the widely available global positioning system (GPS) and geographic information system (GIS) data. However, it still requires significant efforts to process the huge amount of GPS/GIS data to extract the needed curve information. Therefore, there is a need to develop and deploy a cost-effective means for efficiently extracting curve information from the huge amount of GPS and GIS data that are widely available. In addition, there is a need to develop a method to facilitate transportation agencies' practices for determining the HFST installation recommendation, i.e., the start and end points.

1.2 Research Objectives and Tasks

The objective of this is to develop and deploy a cost-effective means for extracting essential curve information, including curve location, radius, PC, and PT, using widely available GPS and GIS data in support of the determination of HFST installation recommendations. This is accomplished by implementing the automatic curve extraction algorithm developed by the authors (Ai Tsai, 2015), developing a tool to facilitate an efficient curve information extraction process, including automated curve identification, QA/QC, and reporting, and further utilizing the extracted PC/PT for HFST installation recommendation (i.e., the start and end points). The specific activities to be performed under each work task are presented below:

Work Task 1: Develop a tool for extracting curve information

In this task, the Georgia Tech research team developed a tool for effectively extracting curve information by implementing the automatic curve extraction algorithm developed by Ai and Tsai (2015). The major subtasks are as follows:

- 1) The research team reviewed the existing curve identification algorithms/tools and summarized the performance of the existing algorithms;
- 2) The research team developed a smart curve identification tool (Smart-CIE) based on the algorithms developed by the authors (Ai and Tsai, 2015). The tool consists of three core components to facilitate an effective curve identification and extraction along a corridor.

Work Task 2: Validate and refine the developed tool using GPS and GIS data collected in Georgia.

In this task, the Georgia Tech research team worked with the Office of Traffic Operations to validate the curve information from the developed tool using the data provided by GDOT and collected by the research team. The major subtasks are as follows:

- 1) The research team worked with the Office of Traffic Operations and the Office of Transportation Data to gather data for validation, including GPS data and GIS data, etc., on the selected sites. In addition, the research team developed a simulated dataset using defined ground truth with various curve types and radiuses for the validation.
- 2) The research team validated the developed tool by comparing the curve information (e.g., curve location, radius, PC, and PT) extracted by the developed tool using GPS and GIS data to the ground truth derived from synthetic data and manual survey data on the actual

roadways. Further study is recommended to validate the curve information (e.g., curve radius, PC, and PT) extracted by different means, such as CARS (Curve Advisory Reporting Service) and Smart-CIE.

Work Task 3: Develop a component for HFST installation recommendation using the extracted PC/PT

In this task, the Georgia Tech research team developed a component, HFST Start/End Points Identification, for HFST installation recommendations. The major subtasks are as follows:

- 1) The Georgia Tech research team reviewed current practices of HFST installation and interviewed the experts on the effective start and end points for HFST installation.
- 1) The research team proposed an extended component in the Smart-CIE tool for transportation agencies to effectively identify the installation boundary for HFST using the extracted PC/PT and the Texas Transportation Institute (TTI) HFST installation guidelines.

Work Task 4: Conduct training on the developed tool and prepare a final report.

In this work task, the Georgia Tech research team prepared training materials and the final report by using GDOT's research project guidance and EDC program guidance. The research team documented, organized, summarized, and disseminated research findings obtained in each work task in this draft final report for review and comment. The research team will incorporate the comments into the final report.

1.3 Organization of This Report

This report is organized as follows. Chapter 1 introduces the background, research need, objectives, and work tasks of this research project. Chapter 2 presents a literature review of the current methods and tools for curve identification. Chapter 3 presents the developed automated curve information extraction tool (i.e., Smart-CIE) and a case study using the actual GPS data to demonstrate that the developed tool provides an effective means for extracting curve information, including curve radius, PC, and PT. Chapter 4 presents the development of an extended component, HFST Start/End Points Identification, for determining the HFST installation boundary (i.e., the start and end points) along with a case study to demonstrate the potential savings over the manual identification method. Chapter 5 concludes the research and provides recommendations for further studies.

2. LITERATURE REVIEW OF CURVE IDENTIFICATION METHODS

Recognizing the traditional field survey methods used by transportation agencies can be time-consuming, labor-intensive and even inaccurate. In previous studies, some researchers have attempted to develop methods that can efficiently and accurately conduct horizontal curve identification and measurement by introducing different data sources and computational methods. With the advances in GIS and the wide availability of the GPS data, the spatial analysis method becomes one of the most feasible, cost-effective options for roadway curve identification and measurement. In this chapter, the current GIS/GPS-based horizontal curve identification and measurement methods are thoroughly surveyed in previous literature. The benefits and challenges of the existing methods are identified. Some other methods, using video log images, satellite images, etc., are also briefly discussed.

2.1 Manual Curve Identification Methods

Researchers started employing a GIS approach to calculating the curve radius from the digital road network (Rasdorf et al., 2012). The chord-offset method is usually applied for computing the curve radius by measuring the chord length of the digitization points and the offset between the chord and the corresponding arc. The interactive interfaces created on the ArcGIS platform, e.g. the Curve Calculator by Environmental Systems Research Institute (ESRI, 2010) and the Curvature Extension by the Florida Department of Transportation (FDOT, 2010), conveniently enable users to select the PC and PT of curves; the corresponding radius will then be computed by automatically extracting the arc and chord length of the curve from a GIS map. As these methods require users to input the PC and PT locations manually, it is challenging to use these

methods for large-scale road networks due to the time-consuming and tedious manual intervention. However, with convenient digitization tools for selecting PC and PT of curves, these methods and tools are still widely used by transportation agencies, although primarily on a project level.

2.2 Automatic Curve Identification Methods

Automatic curve identification methods were further developed to minimize the manual intervention. Instead of requiring users to accurately digitize the PC and PT of curves in the digital road network (e.g., GPS trajectory, GIS map, etc.), automatic methods only need users to input a small number of parameters that are required by the algorithms, and then the algorithms will visit the GPS/GIS data and identify the curves automatically.

Because the primary task for an automatic curve identification algorithm is to differentiate the curved section and the tangent section, the key feature of the heading angle is widely used in previous studies (Camacho-Torregrosa et al., 2015; Drakopoulos and Örnek 2000; Li et al., 2012; Li et al., 2015; Sarbaz et al., 2012; Xu and Wei, 2016; Yun et al., 2013). Within a tangent section, the variance of the heading angles should remain close to zero, whereas, within a curved section, the variation of the heading angles should remain as a non-zero constant. However, as the heading angle is estimated using only a few nearby points within the digital road network (e.g., GPS points, shapefile vertices, etc.), the accuracies of these methods are very sensitive to the data density and data quality. Also, most of these methods are not able to handle compound

curves or spiral curves, except the recent method developed by Camacho-Torregrosa et al. (2015).

Another unique feature of curvature that differentiates the tangent and curved sections is also used in previous studies (Ai and Tsai, 2015; Imran et al., 2006; Rasdorf et al., 2012). Within a tangent section, the curvature should remain close to zero, whereas, within a curved section, the curvature should remain as a non-zero constant. However, the regression-based methods for computing curvature are sensitive to the selection of the input data. Therefore, only a few methods have attempted to use the curvature-based method for curve identification. Similar to the heading-angle-based method, the curvature is estimated using only a few adjacent points within the digital road network (e.g., GPS points, shapefile vertices, etc.) and the accuracies of these methods are very sensitive to the data density and data quality. Thus, most of them are not able to handle compound curves or spiral curves reliably, except the recent method developed by Ai and Tsai (2015). Table 2-1

summarizes the existing automatic curve identification methods using both heading angle and curvature features.

Table 2-1 Summary of automatic curve identification methods

Method	Year	Authors	Curves Types	Data Density	Accuracy
Heading-based	2000	Drakopoulos and Örnek	Simple	<1m	N/A
	2012	Sarbaz et al.	Simple	<1m	90%
	2012	Li et al.	Simple	N/A	97%
	2013	Yun et al.	Simple	<2m	85%
	2015	Camacho-Torregrosa et al.	All types	<1m	N/A
	2016	Xu and Wei	Simple	N/A	96%
Curvature-based	2006	Imran et al.	Simple/Spiral	<1m	Not Reported
	2012	Rasdorf et al. (NHDOT)	Simple	<3m	88%

It can be observed that very few methods have the capacity to identify all types of horizontal curves, while most of these methods demand a relatively high-resolution data frequency (<1 meter). The algorithm developed by Ai and Tsai (2015) shows promising results by identifying horizontal curves of all types with a relatively lower requirement for data frequency (<5 meters). The accuracy of the methods was primarily evaluated based on the percentage of the accurately detected curves by comparing the derived results with the manually established ground truth from field surveys.

2.3 Implementation of Curve Identification Methods

Due to the identified challenges presented in Section 2.1.2, only a few methods attempted by transportation agencies have been found in the previous literature that facilitates identifying and measuring horizontal curves with the intent of minimizing manual interventions. They are the “Curve Extension” developed and implemented by Florida DOT (FDOT) (Rasdorf et al., 2012), the “Curve Finder” developed and implemented by the New Hampshire DOT (NHDOT), and the “CurveFinder” developed by the Wisconsin DOT (WisDOT).

Curvature Extension by FDOT

The “Curvature Extension” is an ArcObjects add-in developed by the FDOT. As pointed out in the previous section, the “Curvature Extension” requires the users to manually input a curve by specifying the PC and PT of each curve. Once the parameters and the data files are defined, and the PC and PT are digitized, the curve radius is determined by creating a circular arc utilizing the chord length, chord angle, and length of the curve along the route. The user has to navigate the data layer manually to find the curve for computation. Figure 2-1 shows the interface of the “Curve Extension.” This tool does not have a batch processing capacity for network-level analysis.

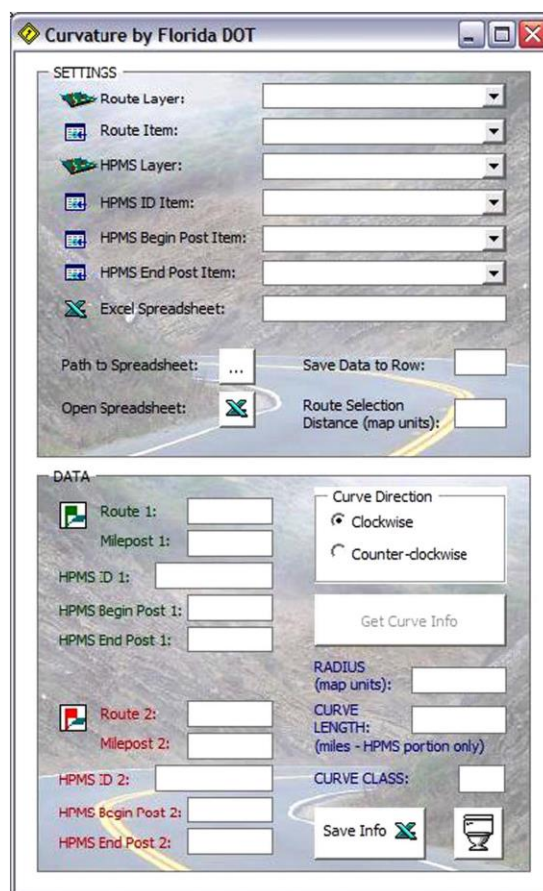


Figure 2-1 The interface for the Curve Extension by FDOT

Curve Finder by NHDOT

The “Curve Finder” by NHDOT (Findley et al., 2012) defines an error quotient that expresses how well a curve is defined. Three consecutive GPS points are used to estimate a small circle, while series of these circles form a circle cluster that can potentially define a curve. The error quotient is calculated by the average of the distance from the center of each circle to the center of the cluster and then normalized by the overall radius. Hence, a zero error means a perfect fit of every circle within the cluster, while a large error indicates that the circles within the cluster are unlikely to form a curve. This method primarily works well on simple curves, but it is very challenging for this method to identify and measure complex curves because the circle estimation uses only three points and is not robust enough to capture the transition from different curved segments within a complex curve. Figure 2-2 shows the interface of the “Curve Finder.”

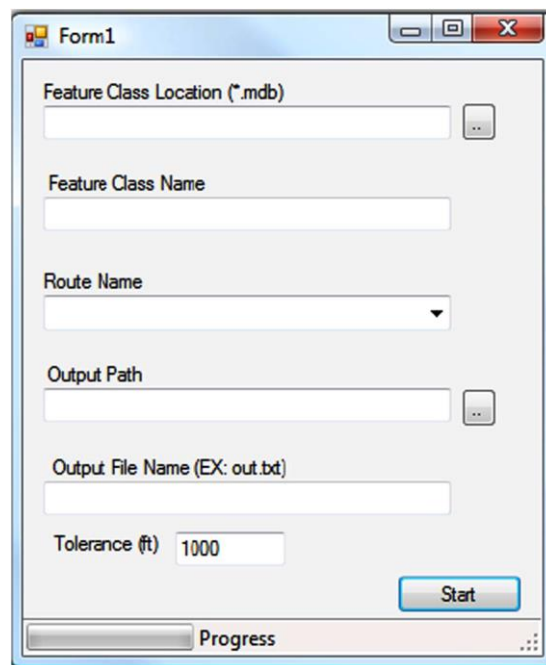
The image shows a Windows-style application window titled "Form1". It contains several input fields and a button. The fields are: "Feature Class Location (*.mdb)" with a browse button, "Feature Class Name" with a text box, "Route Name" with a dropdown menu, "Output Path" with a browse button, "Output File Name (EX: out.txt)" with a text box, and "Tolerance (ft)" with a text box containing the value "1000". A "Start" button is located at the bottom right of the input area. At the bottom of the window is a progress bar labeled "Progress".

Figure 2-2 The interface for the Curve Finder by NHDOT

CurveFinder by WisDOT

The “CurveFinder” by (Li et al., 2012; Li et al., 2015) defines a threshold for the bearing angle to differentiate the curved sections from the tangent sections. The bearing angle is computed consecutively by using the adjacent three points along the roadmap and compared with the calibrated threshold. A corresponding add-in using ArcGIS programming package ArcObjects was developed to facilitate the operation of road layer selection, results in query and selection, etc. Using the selected testing dataset, the “CurveFinder” can correctly identify 96.7% of the tested curves (curve vs. non-curve). However, the “CurveFinder” can only differentiate two curve types, i.e. simple curve and compound curve and only demonstrates a 79% classification rate (simple curve vs. compound curve) due to the limited precision of the bearing angles computed by the consecutive three points. Figure 2-3 shows the interface of the “CurveFinder.”

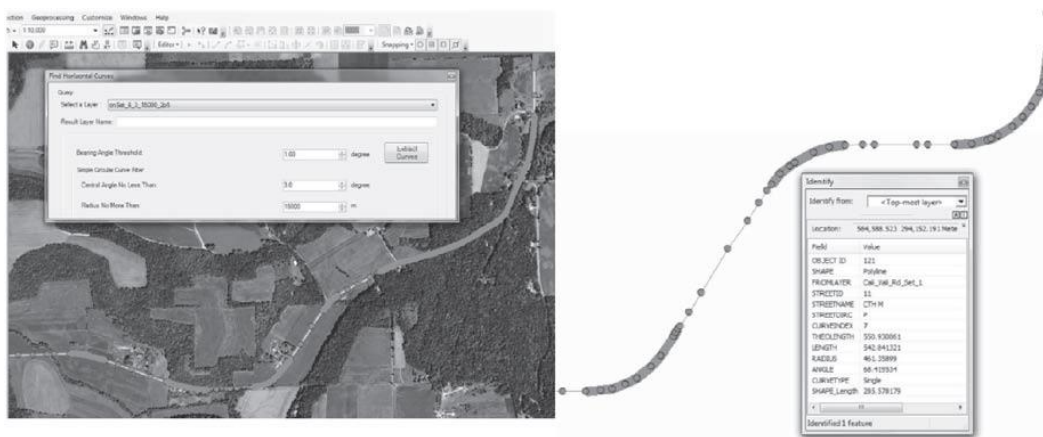


Figure 2-3 The interface for the CurveFinder by WisDOT

2.4 Image-Based Method

Image-based approaches have also been developed using either rectified ortho-imagery (Dong et al., 2007; Easa et al., 2007) or calibrated video log images (Tsai et al., 2010). Easa et al. (2007) also developed a method for extracting horizontal curves using Ikonos satellite imagery. The developed algorithm used a Hough transform to detect curve and tangent sections of horizontal curves from satellite imagery and can be applied to extract both simple and reverse horizontal curves. Dong et al. (2007) developed an approximate algorithm to extract spiral curves from high-resolution satellite imagery using automated extraction methods with a Hough transform to images. Tsai et al. (2010) developed a video-log-image-based method that contains three major steps, including a Canny edge-detection-based method for road extraction, a Hough-transform-based method for curve identification and fitting, and a camera calibration model for curve radius computation. Although some of these image-based methods have shown promising results, the lack of the availability of high-resolution data has hindered further implementation of these methods.

2.5 Summary

Horizontal curve information, including location, radius, and PC and PT, is critical for transportation agencies to support their safety improvement programs. The traditional manual method in the field is time-consuming, tedious, and may be hazardous to field engineers. With the advances and the wide availability of the GPS and GIS data, the GPS/GIS-based method becomes a feasible, cost-effective option for roadway curve identification and measurement. Transportation agencies have implemented several of the existing GPS/GIS-based methods, but

most of these tools are still used for project-level or small-scale network-level analysis due to the combination of the limited algorithm accuracy and the limited functionality and flexibility of the tool. Most of the existing methods either still require extensive manual digitization or still need improvement in identifying and classifying complex curves. Although the GPS/GIS-based methods in previous studies have demonstrated their potential and effectiveness in extracting horizontal curves, it is recognized that a fully automated curve information extraction algorithm is a challenging task, and there is a clear gap between the accuracy of the existing method and the urgency from transportation agencies. Currently, an effective method for extracting curve information from the GPS and GIS data widely available is still lacking. Therefore, there is a need to develop a convenient tool for transportation agencies to efficiently conduct curve information extraction by integrating the state-of-art curve extraction algorithm with a convenient editing toolset. The automated curve extraction algorithm should have the capacity to extract the majority of the curve information correctly (i.e., correct location and PC and PT) and accurately (i.e., accurate radius measurement), while the editing toolset should provide the flexibility to revise the incorrect results and to add new results efficiently. As a fundamental curve extraction platform, such a curve information extraction tool can be further expanded to many other safety improvement programs for transportation agencies, including the start/end point determination of HFST placement.

3. SMART-CIE: AN SMART CURVE INFORMATION EXTRACTION TOOL

The objective of this project is to develop a smart curve information extraction (Smart-CIE) tool to assist transportation agencies in efficiently and accurately extracting detailed curve information (including curve location, radius, PC, and PT) from the huge amount of GPS and GIS data that is widely available. The Smart-CIE implements the automatic curve identification algorithm developed by Ai and Tsai (2015) and incorporates a toolset to facilitate an efficient curve information extraction process. In this chapter, the development of the three core components of the Smart-CIE tool is presented first, followed by the detailed functionality and validation of the component.

3.1 Overview of the Smart-CIE Tool

To provide a convenient means to efficiently identify curve locations along with accurate curve information (radius, PC, and PT) using GPS and GIS data, the Smart-CIE consists of three core components: a) an automated curve identification, b) a map-based quality assurance and quality control (QA/QC) component to incorporate engineers' knowledge, and c) a map-based curve information reporting. With these components, the Smart-CIE enable users to effectively extract curve information along a corridor.

Figure 3-1 shows the operational flow of the three core components and the corresponding main steps within each component. The proposed operational flow was developed to facilitate the adoption of different types of inputs and produce meaningful curve information for

transportation agencies. Following the proposed operational flow, an ArcGIS add-in using the ArcObjects framework and Microsoft Visual Studio C# was developed. The add-in is compatible with the ArcObjects library under the environment of ArcMap 10.3.1. In the following sections, the detailed designs for the three core components are presented.

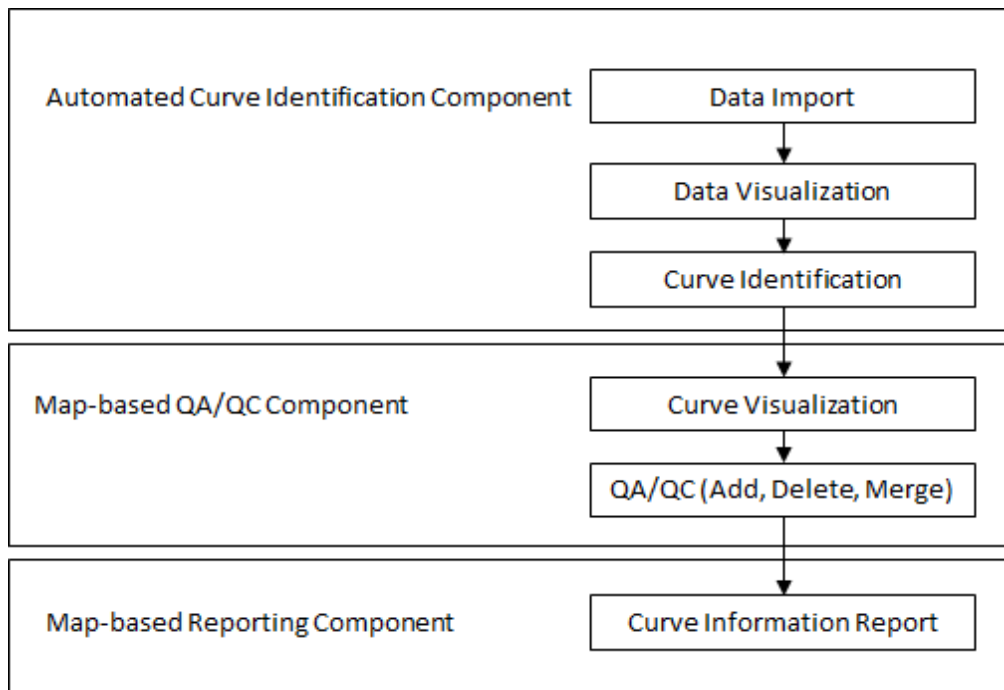


Figure 3-1 Proposed operation flow of the Smart-CIE Tool

To accommodate popular spatial data formats, the research team made additional efforts to allow two types of inputs for the Smart-CIE tool, i.e., comma-separated values (CSV) and shapefile.

The two data formats are described as follows:

- CSV: The CSV file should contain GPS point coordinates in the right sequence and must include three data items: latitude (in decimal degree), longitude (in decimal degree), and point ID (incremental integer). This data format is the common output from the GPS receiver. Typically, latitude and longitude are recorded at each location, and locations are sequentially collected. The sequential information is recorded by the point ID that is critical

for the subsequent processing. The GPS data will be imported and converted into a projected coordinate system based on users' input.

- Shapefile: The shapefile inputs can be either a point or polyline feature class of a certain route. Transportation agencies usually maintain this data format for their roadway network basemaps. Compared to the CSV file format, the shapefile usually contains metadata, such as datum, projection, etc.

Currently, the Smart-CIE tool is designed to process one route at each time. That is, the input data needs to contain one and only one route. For example, any intersected networks need to be segmented into a list of single routes before the Smart-CIE can be applied to process the network one route at a time. More importantly, the input data, especially the CSV file, must be in sequential order. There are a few other factors from the input data that may affect the performance of the Smart-CIE tool, including 1) the input data should not include any cluster points, e.g., GPS points, while the vehicle is stationary; 2) the interval between the sequential data should not be above 15m (i.e., coarse input data); 3) the interval between the sequential data should be as homogeneous as possible to maximize the performance.

3.2 Automated Curve Identification Component

The objective of the automated curve identification component is to batch process the huge amount of the input GPS and GIS data and automatically extract the corresponding curve information, including curve location, radius, PC, and PT. Based on the literature review, it is recognized that a fully automated curve information extraction algorithm with 100% identification rate is an extremely challenging, if not impossible, task. However, the

incorporation of the automated curve identification algorithm has the potential to significantly reduce the manual input for curve extraction operation, especially in a large roadway network. The algorithm developed by the authors of this project is identified as a feasible candidate for curve extraction and implemented in this component. In this section, the principle of the algorithm is presented first, followed by the detailed functionalities and steps designed to seamlessly integrate this algorithm into the Smart-CIE tool.

3.2.1. Algorithm Overview

The core of the automated curve identification component, the algorithm developed by PI (Ai and Tsai, 2015), is briefly introduced in this subsection. The algorithm adapts an iterative process to identify the best-fitted curves along a route trajectory data. In this algorithm, a segment is considered as the minimal unit that has a single radius. A curved segment has a radius of R (a curvature of R^{-1}), while a tangent segment has a radius of infinity (a curvature of zero). A simple horizontal curve consists of curved segments and tangent segments delineated by PC and PT, as shown in Figure 3-2(a). For more complex horizontal curves, there are adjacent curved segments or separated curved segments divided by inner tangent segments, e.g. a reverse curve shown in Figure 3-2(b). The objective of this algorithm is to accurately identify the curved and tangent segments, to reliably cluster the corresponding segments into different horizontal curve types, and to measure the corresponding radii along the identified curve.

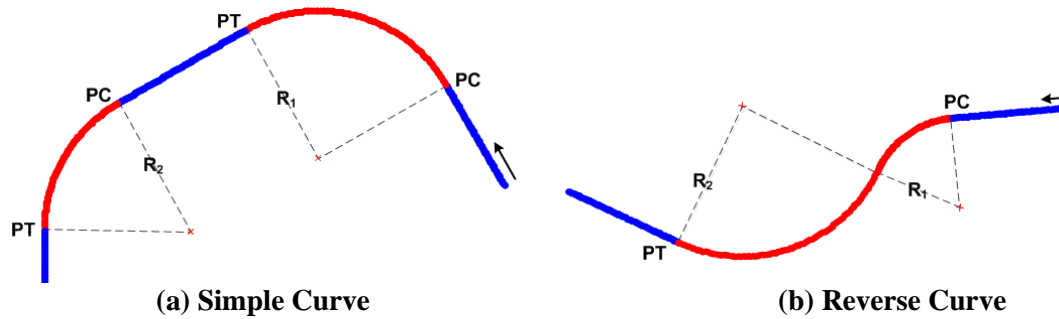


Figure 3-2 Illustration of horizontal curves delineated by PC and PT

The curve identification algorithm of Ai and Tsai (2015) consists of three key steps. Figure 3-2 shows the flowchart of the algorithm. First, the GPS data that represents the roadway is sequentially processed and segmented into delineated segments using an iterative circular fitting. The classic circular fitting method proposed by Kasa (1976) is applied at each iteration. The iterative circular fitting method attempts to find the best fit using an exhaustive search by iteratively increasing the regression size. Instead of selecting a fixed number of neighboring points for fitting, an incremental number of neighboring points are attempted until arriving at the least fitting error.

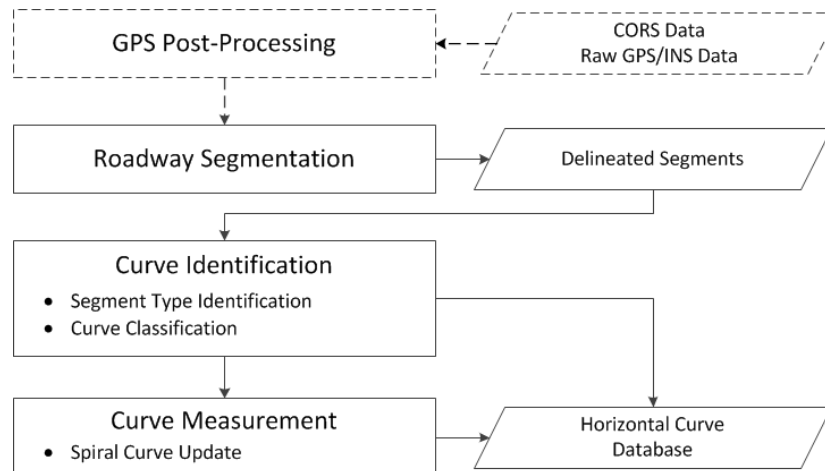


Figure 3-3 Flowchart for the proposed automatic horizontal curve identification and measurement method

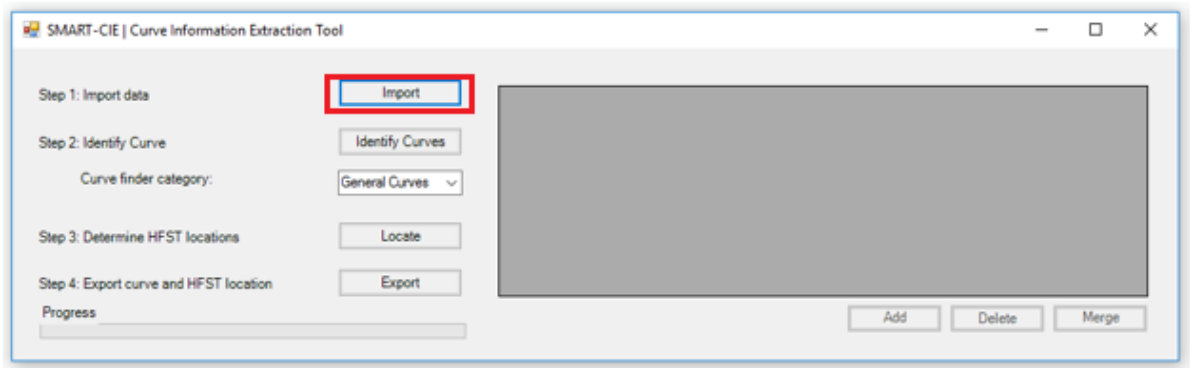
Second, the delineated segments are further clustered into curve sections and classified into different curve types based on their spatial distribution and adjacency. For example, curves are classified as a compound curve if a stepwise pattern is shown for two consecutive curves in the measured curvature diagram. Third, the radii that correspond to each curve are computed. Specifically, the radii along spiral curves are updated using a spiral fitting instead of the aforementioned circular fitting. By updating the curvature measurement at each point at spiral curves in this step, the horizontal curve identification and measurement processes are complete. The detailed algorithm is included in Appendix A.

3.2.2. Operation Flow

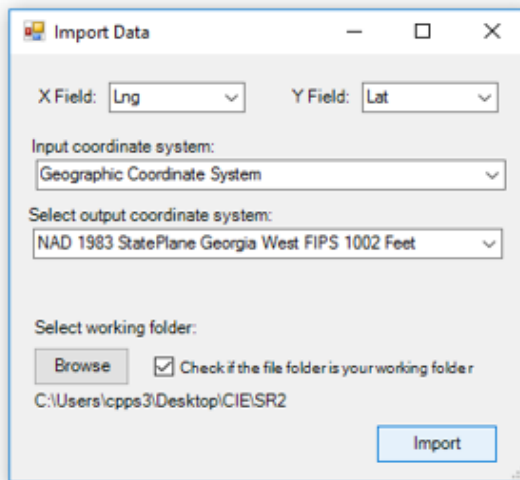
The detailed operation flow of the automated curve identification component includes four primary steps, including data import, data format checking, data visualization, and algorithm execution.

- Data Import: Data import is to process and import input data file into a personal geodatabase so that the subsequent automated curve identification can proceed. The developed tool can currently import two supported data formats (i.e., CSV and shapefile). This step involves 1) obtaining the input data file and other inputs based on the data format, 2) processing and projecting the data into a projected coordinate system if needed, and 3) importing the data into a personal geodatabase.

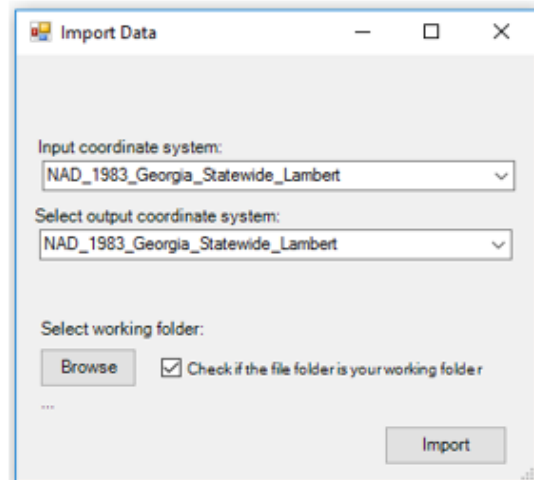
Figure 3-4 (a) shows the user interface for Smart-CIE. The user clicks the “Import” button to start the data import process. Upon clicking the "Import" button, a file browser opens, and the user selects the input file and specifies the necessary inputs based on the data format. For CSV file, the required inputs are the field containing the X-Y coordinates, the input coordinate system (if any), the output coordinate system, and the working folder. As shown in Figure 3-4 (b), the "X Field" and the "Y Field" can be selected from the drop-down menu.



(a) Smart-CIE user interface



(b) Inputs for CSV file



(c) Inputs for shapefile

Figure 3-4 Import data

The output coordinate system is specified to convert the input data into a user-specified projected coordinate system. This is especially required for GPS data, which is often stored in the geographic coordinate system (i.e., longitude and latitude). The curve identification

algorithm calculates distance based on two-dimension surface, and hence the data has to be converted into in a projected coordinate system. The working folder is to store all the intermediate files and outputs. The user must finally click the "Import" button to import the data. For shapefile, the inputs required are the input coordinate system (if needed) and the output coordinate system, as shown in Figure 3-4 (c). The input coordinate system will be retrieved from the shapefile if available. The shapefile can be either point or polyline. For the polyline, an additional step of densification will be applied to densify the input the polyline.

- Data visualization: After the data is imported into the personal geodatabase, it can be displayed on a map with some of the basic information about the input data, e.g., data format, the number of points, distance coverage, projection, etc., as shown in Figure 3-5. The users can visually check the data (e.g., any clustered point) to ensure the data is clean.

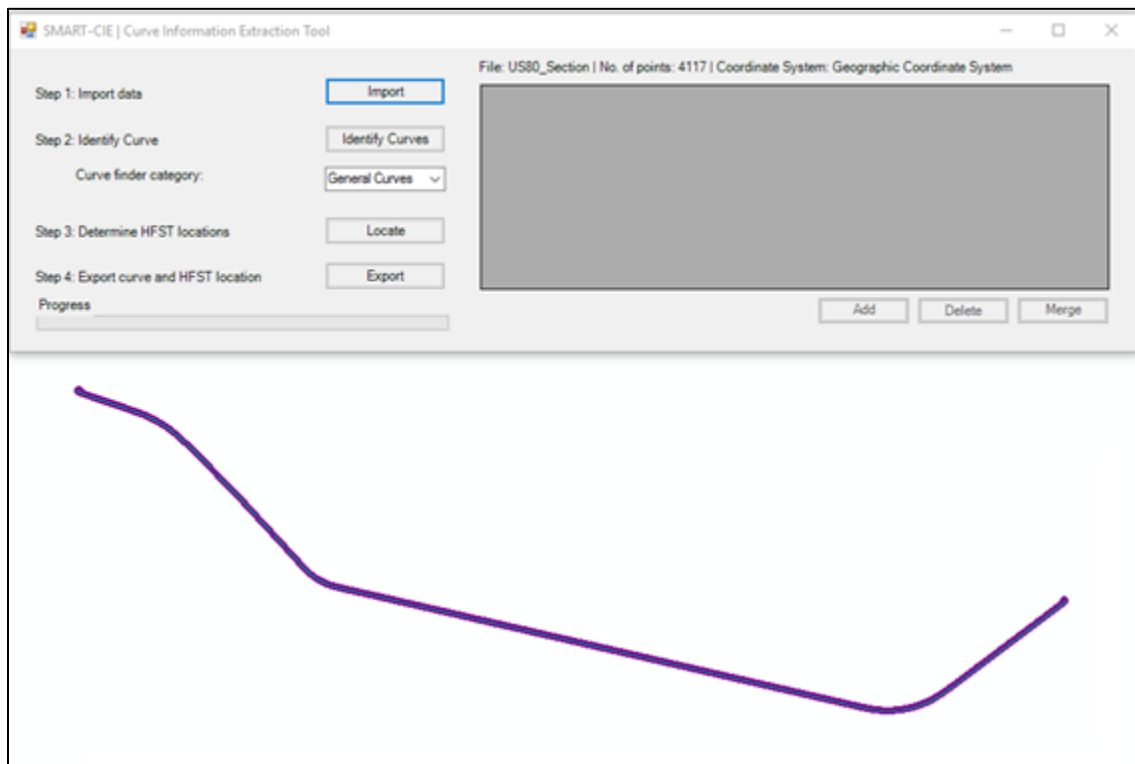


Figure 3-5 An example of input data

Curve Identification: The curve identification algorithm described in Section 3.1 will be applied to the input data. The user may click the "Identify Curves" button to run the algorithm on the roadway segment and display the curves identified on the roadway on the map, as shown in Figure 3-6. The curves are listed in a grid, as shown in Figure 3-6 (a), with the information including Curve ID, status, coordinates of center of curve, curve radius, curve deflection angle, coordinates of PC and PT. The user can also see the curve on the map, as shown in Figure 3-6 (b).

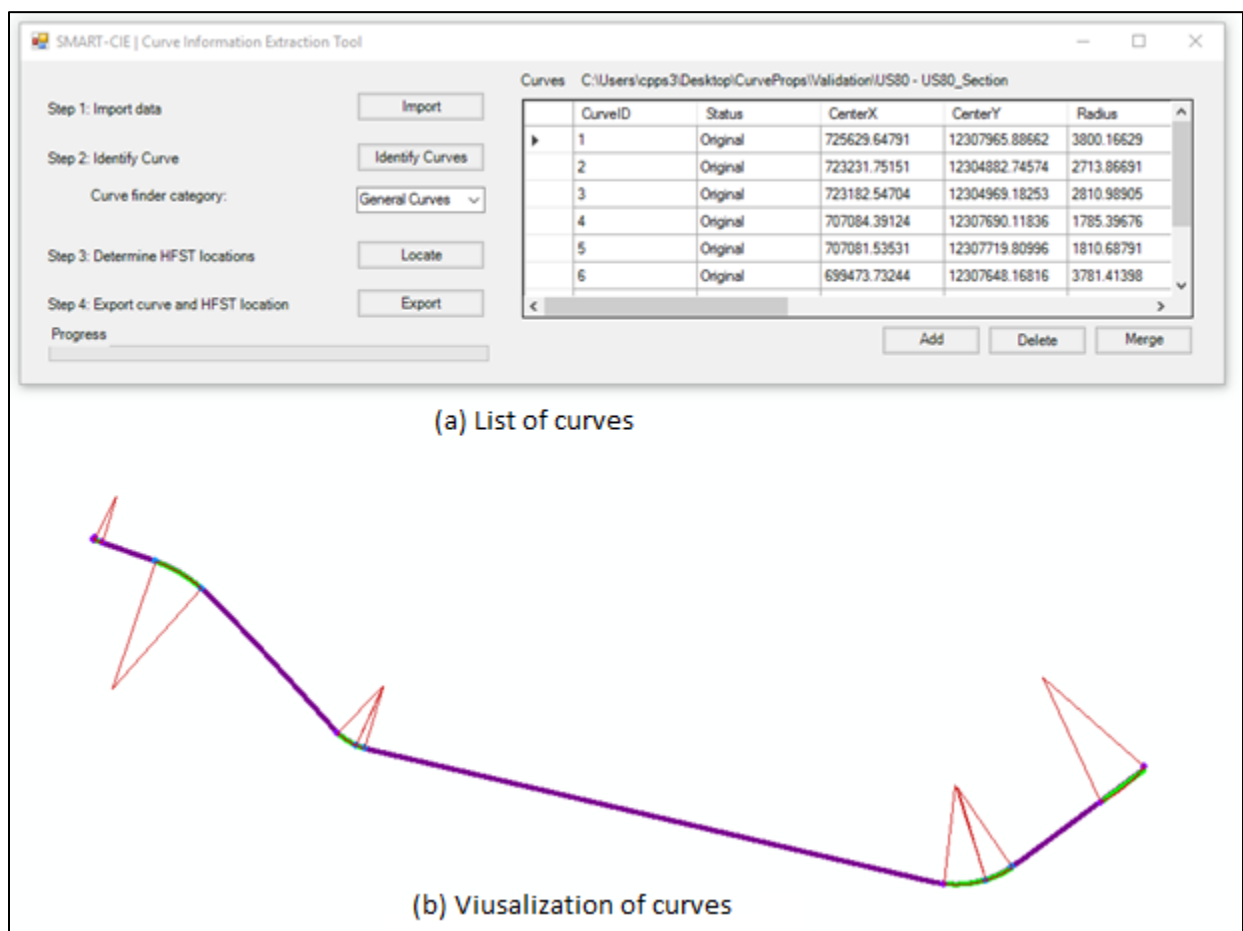


Figure 3-6 An example of the curves identified by the tool

3.2.3. Validation of the Automated Curve Identification

The automated curve identification component was validated using simulation and field tests to quantitatively evaluate its performance. Simulation tests using synthetic curve data were performed first to evaluate the accuracy of the proposed method regarding type identification. Field tests using real-world curve data were then conducted to evaluate the performance of the proposed method regarding measurement by comparing it with the ground truth derived from maps and satellite imagery.

Simulation Tests

The objective of the simulation test is to assess the accuracy of the automated curve identification. Because it is challenging to prepare a comprehensive dataset from the real world with all types of curves, especially the complex curves, a comprehensive synthetic data set was generated based on a series of pre-defined or known curves along with observed GPS disturbances. The curves and tangent sections are created sequentially by inserting synthetic road sections with known geometry formula. The synthetic data is then sampled into discrete synthetic GPS trajectory. To be consistent with the reality, the synthetic dataset is dimensioned to the scale of the real world curves (radius ranging from 100ft. to 4500ft.), and the GPS trajectory is then discretized based on the GPS acquisition frequency of 5Hz and default driving speed of 60mph. Several types of GPS disturbances observed from the real GPS track is a simulation in the synthetic data, including random GPS error, missing GPS points, and local vehicle maneuvering. These GPS disturbances can locally interrupt the continuity of the GPS trajectory. In total, there are 385 curves (including four types of horizontal curves whose radii range from 100ft. to 4500ft.) generated for this simulation test. The accuracy of the simulation test is evaluated by both the detection performance and the classification performance.

For the detection performance, Type 1 (i.e. false negative, a failure to detect an existing curve) and Type 2 (i.e. false positive, a misidentification of a straight section as a curve) errors were used. A Type 1 error is reported when a curve in the ground truth is completely missed or the detected curve section overlaps less than seventy-five percent of the corresponding ground truth, while a Type 2 error is reported when a detected curve section is complete at the straight section in ground truth. It is noted that only 38 Type 1 cases and 27 Type 2 cases were identified, resulting in a correct detection rate of 90.1%. It should also be noted that out of the 38 Type 1 errors, there were only ten curves that were completely missed, while the remaining 28 curves were still partially detected by the proposed method.

For the classification performance, a confusion matrix is used to evaluate how the proposed method distinguishes different curve types. The confusion matrix is derived based on the results from the detection results. A total of 347 curves that were correctly detected were tested for classification. Table 3-1 shows the confusion matrix results among different types. It can be observed from the diagonal cells of the confusion matrix that the proposed method demonstrated overall good performance in differentiating the four curve types. Overall, 87.3% (302 out of 346) of the detected curves were correctly classified by the proposed method. Even for the challenging spiral curves, the proposed method achieved a 63.3% correct classification rate.

Table 3-1 Confusion matrix for curve classification test

Proposed Method Ground Truth	Simple Curve	Compound Curve	Reverse Curve	Spiral Curve
Simple Curve (117)	90.6%	6.0%	0.0%	3.4%
Compound Curve (103)	5.8%	86.4%	0.0%	7.8%
Reverse Curve (78)	1.3%	0.0%	98.7%	0.0%
Spiral Curve (49)	24.5%	12.2%	0.0%	63.3%

However, 36.7% of the spiral curves were incorrectly classified as simple or compound curves due to the incorrect delineation between the center curve and its corresponding spirals. If both the forward and back spirals are incorrectly delineated, a misclassified simple curve will be reported, while if either of the spirals is incorrectly delineated, a misclassified compound curve will be reported.

Field Tests

The objective of the field test is to evaluate the performance of the automated curve identification method in curve measurement. Three testing datasets were collected, including 25 simple curves with different radii. The corresponding GPS data was collected at 5Hz (approximately 5m interval) on these three roads. Figure 3-7 shows the map of the selected roadways. The Georgia Tech Savannah campus contains consistent curve radius on a circular loop road (i.e. five radii ranging from 968 ft. to 1024 ft.). Jimmy Deloach Parkway is a minor arterial with a majority of large and smooth curves (i.e. ten radii ranging from 1409 ft. to 5596 ft.). Industrial Park Road is a local road with a majority of sharp curves (i.e. ten radii ranging between 218ft. to 801ft.).

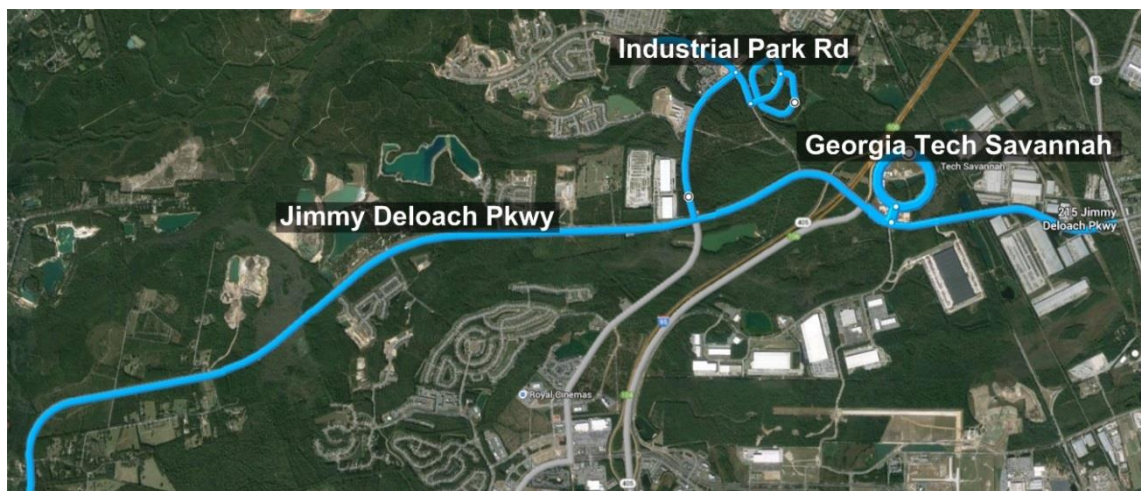


Figure 3-7 Selected roadway in the field test

The satellite imagery for the selected testing datasets was used to digitize the ground truth using ERDAS and MATLAB and measure the accurate curve radius using AutoCAD. The automated curve identification method was then applied to the three testing datasets. All of the 25 simple curves were correctly detected, and the corresponding radius was automatically measured by the proposed method. Figure 3-8 shows the scatter plot of the radius measurements derived from the proposed method against the ground truth measurements derived from AutoCAD. A linear regression with zero intercepts was performed to demonstrate the correlation of the two results, shown as the solid line in Figure 3-8. The slope of 1.0127 and the R^2 of 0.9999 strongly suggest an accurate measurement using the proposed method.

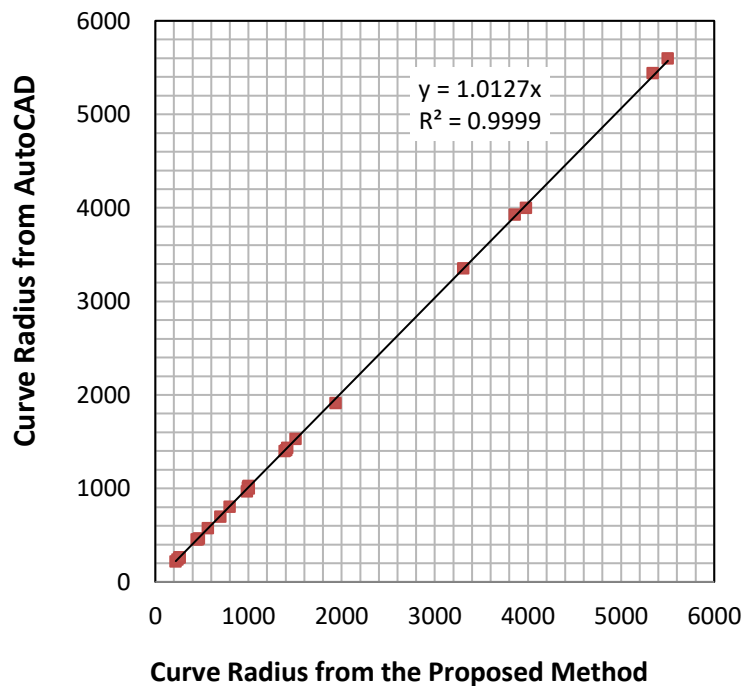


Figure 3-8 Curve radius measurement results of the field test

3.3 Map-Based QA/QC Component

The objective of the map-based QA/QC component is to provide a set of convenient functionalities for the users to edit the identified curves from the automated curve identification component so that a complete set of horizontal curves and their corresponding information can be extracted reliably and efficiently. While the algorithm incorporated in the Smart-CIE tool showed accurate curve identification results, the PC and PT of some curves could not be accurately extracted using the input roadway data, and some of the curves were overlooked by the implemented algorithm. Therefore, a series of editing functionalities, including add, delete, and merge, were developed in this QA/QC component so that the Smart-CIE tool can be adopted for accurate, network-level analysis. This component contains two steps, including the result visualization and results in QA/QC. The key functionalities are presented in the following discussion (the detailed operation procedure is provided in Appendix D):

- **Curve Visualization**: The result derived from automated curve identification will be displayed in two formats, including a map and a table. In the map, the arc of each detected curve and the corresponding PC and PT will be highlighted on the map; in the table, the detailed geometry curve information, including PC, PT, curve radius, and deflection angle, is formatted and populated. The navigation functionality was developed to facilitate convenient navigation of the identified curves using the Smart-CIE tool.
- **QA/QC**: After the results are displayed on a map and in the corresponding table, the user will be provided with editing tools to perform QA/QC operation. The tools will allow users to update the map by adding, deleting or merging curves when appropriate. The tool will also allow the user to update the table by adding, deleting, or updating the curve information. The

operation will automatically update the corresponding information on the map and in the table.

- **Add:** The “Add” function allows the users to add new curves to the roadway that the algorithm did not find. It is a two-step process. It requires the user click two points on the map along the roadway where a curve is expected, as shown in Figure 3-9 (a). This functionality will process all the data between the two clicks along the trajectory and identify the best-fitted curve within the range. The new curve will be added into the resulting database and subsequently visualized on the map, as shown in Figure 3-9 (b).

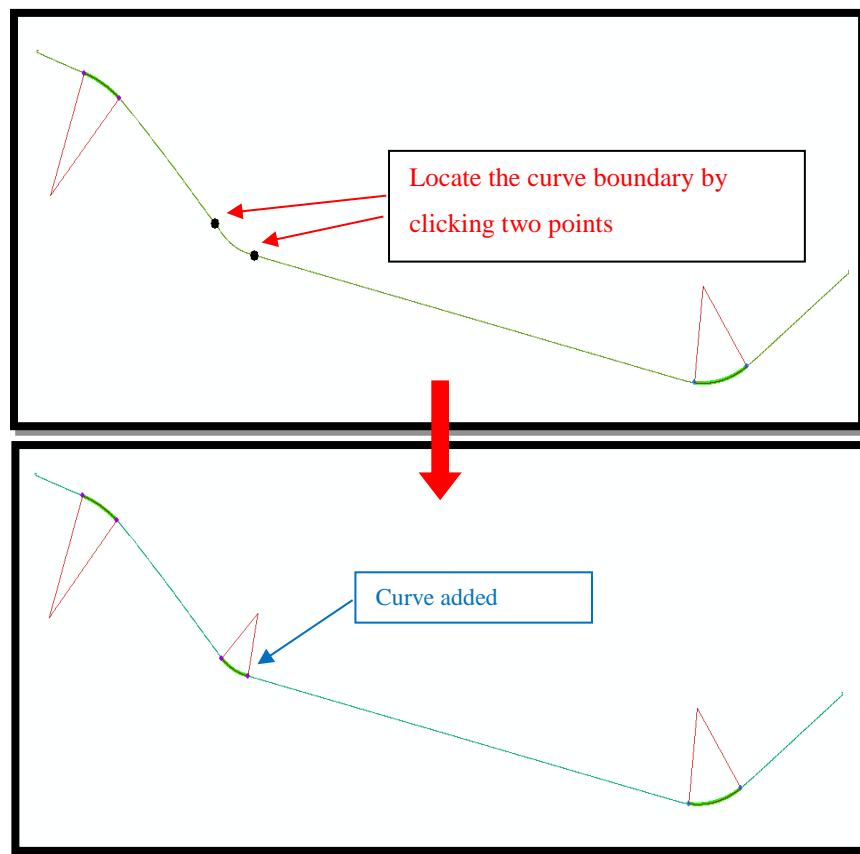


Figure 3-9 The operation of the Add functionality for the Smart-CIE tool

- **Delete:** The “Delete” function allows the users to delete the curves that are incorrectly identified by the Smart-CIE tool or by other users. These curves can be selected by clicking on the map as shown in Figure 3-10 (a) or by checking the tabulate record in the database. By confirming the delete operation, the undesirable curves will be removed from the results, as shown in Figure 3-10 (b).

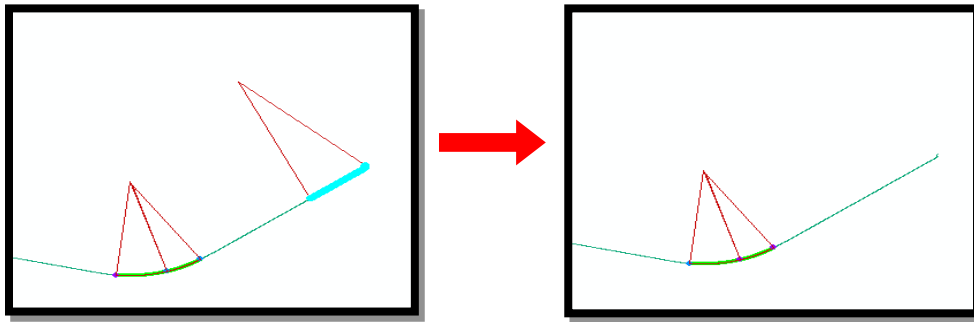


Figure 3-10 The operation of the Delete functionality for the Smart-CIE tool

- **Merge:** The "Merge" function helps the users to combine the identified curves that are a part of the larger curve on the roadway. These curves can be selected by clicking on the map as shown in Figure 3-11 (a) or by checking the tabulate record in the database. By confirming the merge operation, the selected curves will be combined into a single curve, as shown in Figure 3-11 (b). The corresponding geometry information will be updated based on the new boundary of the merged curve.

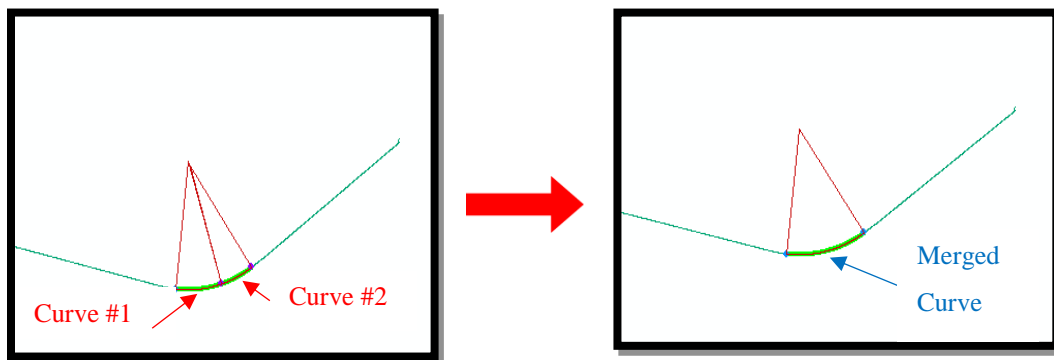


Figure 3-11 The operation of the Merge functionality for the Smart-CIE tool

3.4 Map-Based Curve Information Reporting Component

The objective of the map-base curve information reporting component is to provide a convenient means to export the extracted curve information. In this study, the results of the Smart-CIE tool are reported in a tabulate format listing all the extracted curves in the processed network and their corresponding curve information, including location, radius, and PC and PT. Convenient export functionality was developed in the ArcGIS environment to export the curve information stored in the geodatabase into a universally readable Excel file. The export functionality can also be extended to support agencies' curve safety improvement program in other data formats, e.g., SafetyAnalyst, model inventory of roadway elements (MIRE), etc.

3.5 Case Study on State Route 2

A case study using GPS data collected on State Route 2 in Georgia was conducted to demonstrate the performance of the developed Smart-CIE tool and its feasibility in efficiently and accurately extracting curve information, including curve location, radius, and PC and PT. A map of State Route 2 in the study area is shown in Figure 3-12.

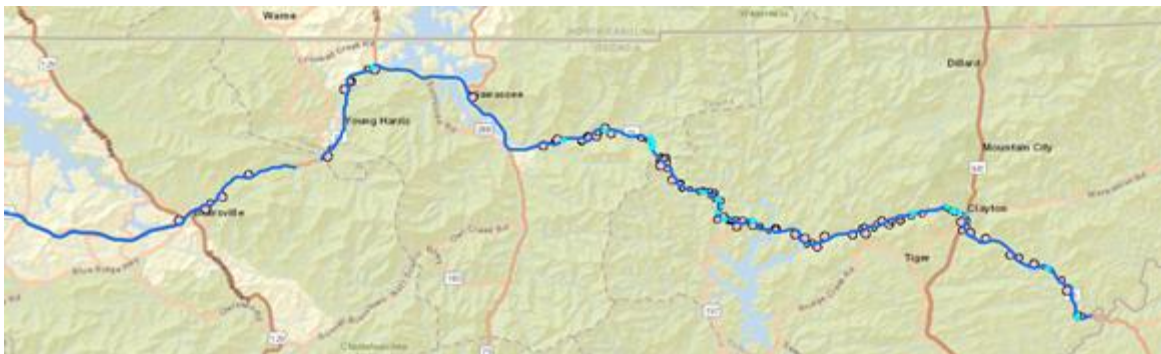


Figure 3-12 Overview of the case study roadway section on State Route 2

State Route 2 in Towns County and Rabun County, consisting 92 curves in approximately 60 centerline miles, was selected for the case study because it covers various curves with a wide range of types (e.g., simple curve, reverse curve, and compound curve) and radii. The data collection, processing procedure, and the results are presented in the following.

3.5.1 Data Collection

GPS data on State Route 2 in Towns County and Rabun County was collected on March 25th, 2016, using the Georgia Tech Sensing Van (GTSV) at 5 Hz frequency at the posted speeds (approximately 35-45 mph) along the route. Figure 3-13 shows a picture of the GTSV and an example of the collected GPS trajectory data. In this study, the raw GPS trajectory data is acquired at 5 Hz frequency at the posted speeds (approximately 35-45 mph). The data was collected in two directions, but only the southbound data was used in this case study.



Figure 3-13 The data collection vehicle and the sample GPS trajectory data

A total of 11,773 GPS points were collected in the southbound lane and used for identifying the curves on the route. Because the data is collected through a mobile platform, many factors will impact the accuracy of the raw GPS trajectory data, e.g. the dilution of precision (DOP), temporary satellite loss, limited dwell time, etc. Therefore, the raw GPS trajectory data were post-processed through differential adjustment and triangulation using base station data collected

by the continuous operating reference stations (CORS) to reduce the impact of these negative factors on the accuracy. In this study, Applanix POSPac MMS software was used to post-process the GPS data.

3.5.2 Data Processing Procedure

To facilitate a convenient navigation of the curve results by the Smart-CIE, a data processing procedure was recommended using the developed Smart-CIE tool, as presented in previous sections. Figure 3-14 shows the interface for the curve identification procedure.

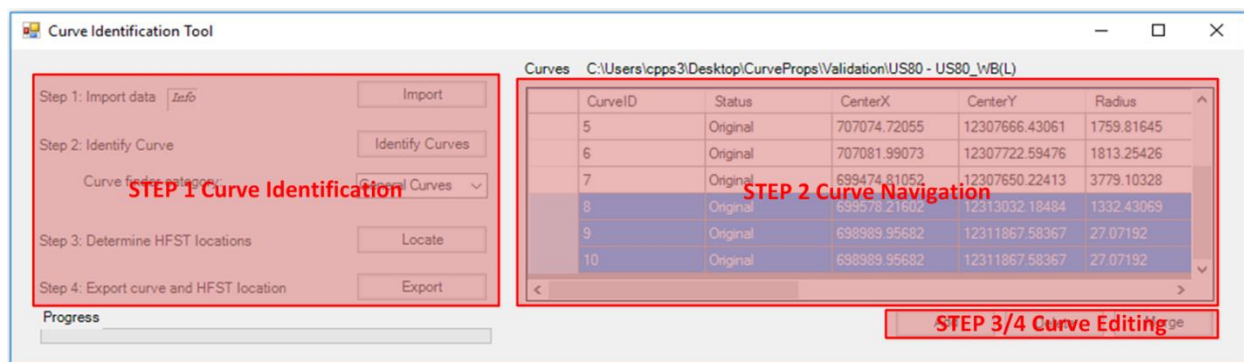


Figure 3-14 The QA/QC interface for Smart-CIE in network-level analysis

- **STEP 1 Curve Identification.** By importing a network data (e.g., GPS trajectory, GIS shapefile, or KML file), this step will execute the core algorithm in the Smart-CIE and populate the results in both tabulate and map formats. The table will list all the basic information for the automatically detected curves, including PC, PT, radius, deflection angle, etc., while the corresponding results will be visualized on the map.
- **STEP 2 Curve Navigation.** The curve results in the tabulate formation will be the primary navigation panel for the user to conduct QA/QC. The user can sequentially go up or down the

record of the detected curve results. The map will automatically zoom into the currently identified curve and scaled to the full extent.

- **STEP 3 Curve Editing – Add, Delete and Merge.** By judging from the zoomed-in curve record in the map, the user can determine if further editing operation is needed or not, including adding, deleting, and merging as presented in the previous sub-section. Navigating the detected curve record will not work for the curves omitted by the Smart-CIE tool. Therefore, navigation of the complete network is necessary. Therefore, in this step, it is recommended that the user navigate the entire input network in the map frame and identify any curves that may be missed by the Smart-CIE tool. As the entire network will need to be reviewed, a playback functionality is developed to navigate all the data points within the network so that any missed curves can be identified through the user's visual inspection. At any point of the navigation, a new curve can be added using the Add tool presented in the previous subsection.

3.5.3 Curve Inventory Results

The processing time for the 31 centerline miles is less than 20 minutes (including an algorithm processing time of less than 30 seconds and the manual intervention time, which is approximately 19 minutes). More than 86% (80 out of 92 curves) of the curves were correctly extracted by the Smart-CIE, and the remaining 12 curves were also conveniently inventoried using the interactive tool (e.g., add, delete, and merge curves). All of the undetected curves include more than three consecutive reverse curves that may confuse the algorithm with relatively small regression errors. Figure 3-15 (a) shows an example of the undetected curves. A set of optimized parameters were identified for different data sources (with different data density and accuracy) so that minimum manual intervention is needed. In addition, some of the curves

were detected with incorrect PCs and PTs. Figure 3-15 (b) shows an example of the undetected curves (in the green bounding box). Therefore, the manual editing was applied subsequently to correct these results. The incorrect identification of PCs and PTs occurs primarily when a curve is tight and with a small deflection angle, where the iterative circular regression in the algorithm may include more non-curve related points to converge at a smaller regression error.

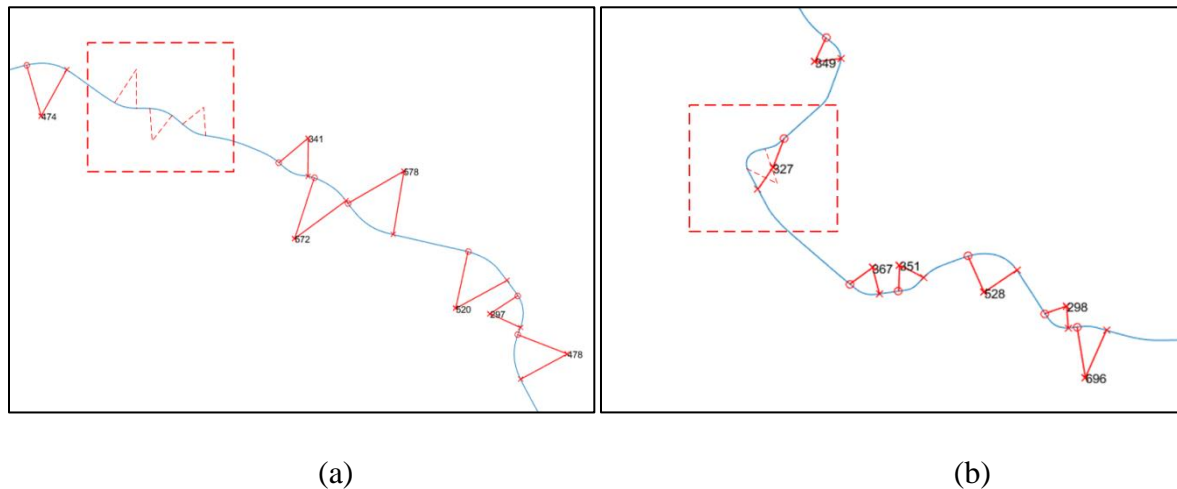


Figure 3-15 Examples of undetected curves and incorrectly detected curves

It is observed that the automated curve identification component of the Smart-CIE tool successfully reduces the majority of the manual effort for curve identification and information extraction. Meanwhile, as the integrate algorithm still produces undetected or incorrectly detected results, the map-based QA/QC component and reporting component of the Smart-CIE tool, i.e., the interactive editing functionality and visualization of the results, provide a convenient means for the users to update and correct the remainder of the undetected or incorrectly detected curves, respectively. The Smart-CIE provides results that create an accurate and efficient means for transportation agencies to extract curve information on a network level, which will provide essential data for the agencies to use in developing their curve safety improvement programs (e.g., SafetyAnalyst, MIRE, etc.). This statewide curve inventory also

includes the improvement of current segmentation-based HFST site selection program to a curve-based HFST site selection program which is more accurate.

3.6. Summary

A tool (the Smart-CIE) was developed for extracting curve information by implementing the algorithm to extract the corresponding curve information, including location, radius, and PC and PT, using widely available GPS data and GIS data. The Smart-CIE tool consists of three core components, including a) an automated curve identification component, b) a map-based QA/QC component to incorporate engineers' knowledge, and c) a map-based curve information reporting component. Simulation and field tests were conducted to validate the incorporated algorithms of the Smart-CIE tool. The results show that the developed tool has promise and is feasible for supporting efficient and reliable curve information extraction work for transportation agencies. While the embedded algorithm developed by the authors will dramatically improve the productivity of the curve extraction task, other functionalities (e.g., data visualization, curve editing, etc.) will provide a more practical toolset for transportation agencies to efficiently and reliably identify curve locations and extract critical curve information (e.g., curve location, radius, PC, and PT). A case study on State Route 2 was conducted to demonstrate the overall performance of the developed Smart-CIE tool and its feasibility for efficiently and accurately extracting curve information, including curve location, radius, PC, and PT.

4. SMART-CIE EXTENDED COMPONENT: HFST START/END POINTS IDENTIFICATION FOR HFST INSTALLATION RECOMMENDATION

The developed Smart-CIE tool, consisting of three core components, has provided a convenient means for transportation agencies to extract their horizontal curve information in a reliable and efficient manner. In this chapter, an extended component is presented to demonstrate the capacity of the developed Smart-CIE tool that can be implemented for transportation agencies to enhance their current safety improvement programs immediately. By integrating the recommended Texas Transportation Institute (TTI) HFST installation guidelines with the developed Smart-CIE tool, the research team developed an extended component (HFST Start/End Points Identification) for transportation agencies to effectively determine the start/end points of HFST for each curve. This chapter first briefly discussed the installation guidelines recommended by TTI and then followed up with its corresponding implementation. A case study on State Route 2 was conducted using the developed extended component to demonstrate the potential savings that the Smart-CIE tool can bring to accurately and conveniently determining the start/end points of HFST installation.

4.1. Identification of the Start/End Points of HFST

As HFST is a relatively new method for horizontal curve safety improvement, there are very limited number of studies and guidelines that suggest the HFST installation boundary, i.e., the start/end points, to cost-effectively reduce run-off-road (ROR) crash rates. Through a thorough literature search, only one guideline was identified that recommended the HFST installation

boundary, which was developed by DBi and TTI in 2012. The research team further pointed out that most of the current HFST installation projects have been based on this guideline.

In this guideline, it was asserted that HFSTs should be applied to the approach tangent before curves to increase the friction for braking before the curve instead of only being applied to the curved segment. The research team at TTI considered that many ROR crashes might be prevented by extending the HFST from the PC back to the critical point where the deceleration must begin in advance of the curve. The tangent segment before the PC is considered as the responding segment that demands vehicles decelerate so that they can reach a proper speed for navigating the curve before the PC. The research teams at TTI considered that majority of drivers apply their brakes ahead of the critical point. However, the drivers who aggressively decelerate at the last moment will demand more friction and may bear a higher risk of ROR crash, especially when the friction condition of the road surface is degraded (e.g., wet surfaces) or the driver's car has worn tires, etc. In their analysis, a deceleration rate of 10 ft. /s^2 , as a conservative value based on driver comfort, was employed to determine the starting point of HFST application before the PC of the curve. Based on such an assumption, Table 4-1 was proposed by the research team at TTI for looking up the starting point of the HFST for a specific horizontal curve on the PC.

Table 4-1 Look up table for determining the HFST starting point before PC

Approach Speed (mph)	Curve Speed (mph)						
	30	35	40	45	50	55	60
35	35	-	-	-	-	-	-
40	76	41	-	-	-	-	-
45	122	86	46	-	-	-	-
50	173	138	97	51	-	-	-
55	230	194	154	108	57	-	-
60	292	257	216	170	119	62	-
65	359	324	284	238	186	130	68

Figure 4-1 illustrates the procedure for determining the starting and ending points of the HFST installation based on the recommendations provided by TTI. It can be observed that accurately identifying the PC and PT becomes the critical step in determining the HFST installation boundary. Therefore, the developed Smart-CIE tool helps transportation agencies conveniently identify the critical information of the PC and PT's position and then, based on the posted speed and advisory speed, identify the recommended HFST installation boundary. The posted speed and the advisory speed can be input by the safety engineers or can be automatically extracted using video log images (Ai and Tsai, 2012; Ai and Tsai, 2014; Ai and Tsai, 2014; Tsai et al., 2009).

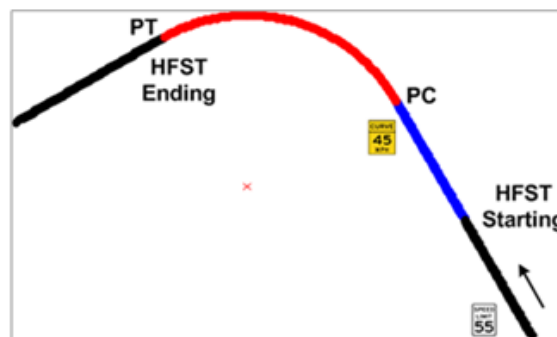


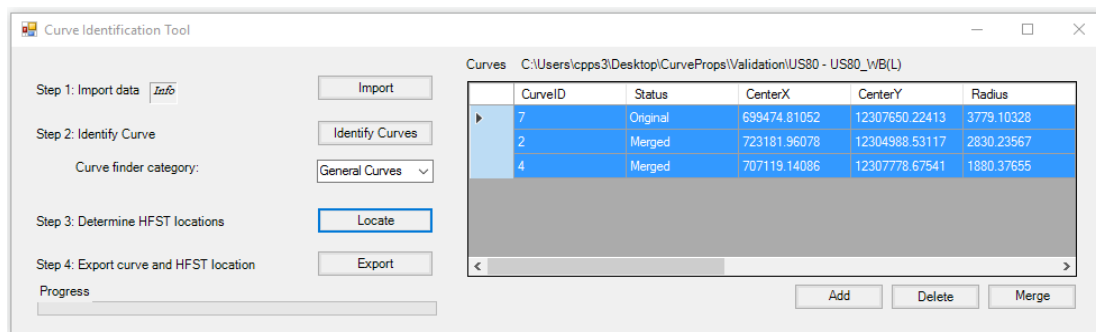
Figure 4-1 Procedure of determining the HFST installation boundary in a curve

4.2. HFST Start/End Points Identification Component

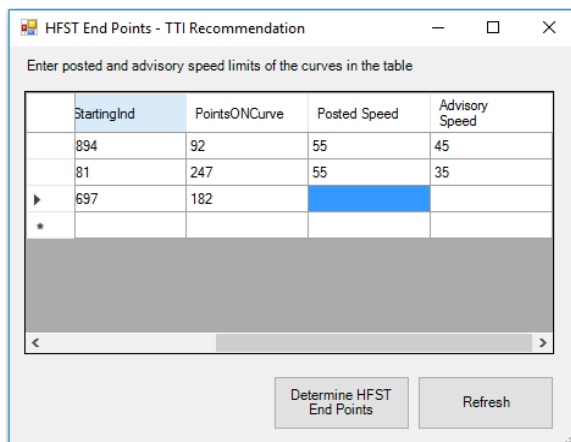
The HFST start/end points identification component was implemented in a commend button in the Smart-CIE. After the curve information is extracted, the user can select a curve (as shown in Figure 4-2 (a)) and input the speed of the curve (as shown in Figure 4-2 (b) to compute the start and end points (x-y coordinates) for HFST installation. Besides the coordinates for the start/end

points, the length and quantity (in square yards) are also computed for the cost estimates. Figure 4-2 (c) shows an example of determining start/end points of HFST.

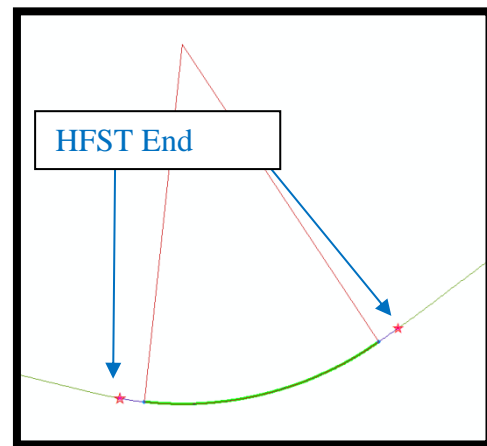
With this information, a smart phone app is recommended for location the start and end points in the field. The information can be exported and loaded onto a smart phone, so the project engineer(s) and/or contractor(s) can find the start and end points productively and precisely by using the navigation system. In addition, this will provide seamless communication in terms of HFST locations between the transportation agencies and the contractors.



(a)



(b)



(c)

Figure 4-2 An example of determining the start/end points of HFST

4.3. Case Study on State Route 2

Before this study, GDOT let a safety improvement project (Project Identification Number, PI #0009993) to improve horizontal safety using HFST, including more than 60 curves on State Route 2 in Rabun County and Towns County. The horizontal curve that is proposed for the HFST in this project was determined by the historical run-off-road crash data analysis and in-field ball bank indicator (BBI) measurement (for the detailed procedure, please refer to research report RP15-04). Field engineers in GDOT further investigated each selected horizontal curve candidate for HFST and determined the detailed starting and ending points of the HFST following the procedure presented in Section 5.1. However, the PC and PT for each curve were determined in the field by the judgment of the field engineers. Figure 4-3 shows an example of the HFST starting point for Curve 7 on State Route 2 in Rabun County determined by using a linear referencing system (i.e., “2.361 – BEGIN HIGH FRICTION SURFACE”).

2.357 Right	ADD W1-5 ADVISORY TURN SIGN AND W13-1P ADVISORY SPEED PLAQUE (35 MPH ADVISORY SPEED)	24.00
2.361	BEGIN HIGH FRICTION SURFACING	24.00
2.380 Right	BEGIN W1-8 CHEVRON ALIGNMENT SIGN PLACEMENT (BOTH DIRECTIONS, SPACED 120 FT)	24.00
2.420 Right	END W1-8 CHEVRON ALIGNMENT SIGN PLACEMENT	24.00

Figure 4-3 Example of the HFST installation boundary on State Route 2 in PI#0009993

The detailed information in PI #0009993 provided a good opportunity for the research team to compare the current HFST installation boundaries with the locations using the developed curve identification tool. The research team collected the detailed sensing data for State Route 2 in Rabun County using the Georgia Tech Sensing Vehicle (GTSV). The GTSV was sponsored by the US DOT and GDOT. It consists of four high-resolution video cameras (at 2448x2048 resolution) for generating a panoramic view of the roadway and the detailed downward view of

the pavement. All the video cameras are synchronized with a high-accuracy GPS/IMU system at 100Hz and geometrically calibrated so that the features extracted from images can be geo-referenced with accurate position information. Using the collected sensing data, the roadway data was acquired using 100Hz GPS in the GTSV, in the X, Y, Z format, while the actual locations (i.e., starting and ending points) of the HFST for each horizontal curve were extracted using the high-resolution cameras.

For each horizontal curve, the comparison between the identified HFST locations in PI#0009993 by GDOT and the HFST locations determined by the developed curve identification tool can be conducted spatially. Figure 4-4 shows an example of the spatial comparison for Curve #47 on State Route 2 in Rabun County. The blue line indicates the GPS trajectory of the roadway, while the green-dashed line shows the coverage of the actual curve arcs that construct this reverse curve. The bold red line shows the starting and ending points of the HFST by GDOT (i.e., S_{GDOT} and E_{GDOT}), and the bold gray line shows the starting and ending points using the curve identification tool by the research team (i.e., S_{GT} and E_{GT}).

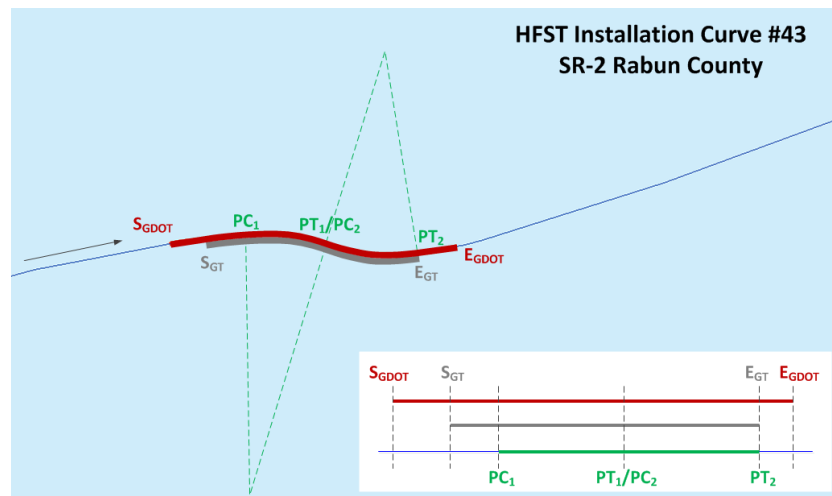


Figure 4-4 Examples of the identified curves with measurements on the State Route 2

The schematic diagram on the bottom right of Figure 4-4 shows the HFST locations identified. It can be observed that, although both GDOT and the research followed the procedures proposed by TTI in Section 4.1, GDOT's coverage of the HFST is larger than the research team's because of the curve location (i.e., PC and PT) identification accuracy. In this case, the difference between the coverage by GDOT and by the research team is more than 700 sq. yd. in each driving direction.

In PI#0009993, there are 57 horizontal curves that were selected for HFST; they cover more than 102,200 sq. yd. of the curved roadway sections. Based on the GDOT's installation location selection in PI#0009993, a total of 140,090 sq. yd. of HFST will be installed in these sections. In comparison, following the same installation location selection procedure developed by TTI, the research team found that if the PC and PT locations can be accurately determined, a total of 116,300 sq. yd. of HFST will be needed, which may potentially save almost 25,000 sq. yd. of HFST material and operation (potentially \$600,000 of saving for this project).

The current practice in GDOT for identifying the locations of HFST is based on the engineers' judgment for determining the PC and PT of a horizontal curve and then follows the procedure developed by TTI. Although the current practice may provide a good estimate for HFST installation, it may misestimate the coverage of the HFST installation due to the error, subjectivity, and inconsistency in identifying the PC and PT in the field by the engineers. The developed curve identification tool provides a convenient, accurate means to identify the PC and PT of each curve and then determines the starting and ending points of the HFST installation following the same procedure developed by TTI. In this particular case on State Route 2 in Rabun and Towns Counties, more than 25,000 sq. yd. of HFST material and operation costs may be saved without compromising performance.

5. CONCLUSIONS AND RECOMMENDATIONS

Transportation agencies have been seeking a cost-effective method for extracting curve information, including curve location, radius, the point of curve (PC), and the point of tangent (PT) by leveraging the GPS and GIS data that is widely available now; such information is much needed but is often lacking. While this approach is promising, it requires significant efforts to extract the curve information from the huge amount of data. There is a need to develop and deploy a cost-effective means for extracting essential curve information, including curve location, radius, PC, and PT, using widely available GPS and GIS data to support the determination of HFST installation recommendations. In this project, a Smart Curve Information Extraction (Smart-CIE) tool was developed and deployed to extract curve information using GPS and GIS data in a reliable and efficient manner by implementing an algorithm developed by the authors to automatically extract curve information from the huge amount of available data. In addition, an HFST Start/End Points Identification component was developed to accurately determine the start and end points for HFST installation; this method uses the PC/PT extracted by the Smart-CIE and the HFST installation guidelines developed by the Texas Transportation Institute (TTI). The outcomes of this study are summarized as follows:

- A smart curve information extraction (Smart-CIE) tool was developed to cost-effectively extract detailed curve information, including curve location, radius, PC, and PT, using widely available GPS and GIS data. The Smart-CIE tool was developed in ArcGIS 10.3.1 as an add-in to streamline the curve identification process. The Smart-CIE consists of three core components, including a) an automated curve identification component that implements the algorithm developed by the authors to automatically extract curve information from the huge amount of available GPS and GIS data, b) a map-based quality assurance and quality control

(QA/QC) component to incorporate engineers' knowledge, and c) a map-based curve information reporting component.

- Validation using simulated and field data collected by the research team was conducted to critically evaluate the performance of the automated curve identification component of the Smart-CIE tool. The validation data contains 385 simulated curves with four curve types (simple, compound, reverse and spiral) and 25 actual curves of the simple curve type. The results have demonstrated the accuracy of the algorithm (i.e., the correlation slope of 1.01 and the R^2 of 0.99 strongly suggest an accurate measurement) previously developed by the authors; its use results in significant improvement in the productivity of the curve information identification process.
- A case study on State Route 2 was conducted to demonstrate the Smart-CIE tool has the capability of conducting a network-level curve inventory efficiently. The tested section covers 31 miles of State Route 2 and consists of 92 horizontal curves. The processing time of the 31 survey miles (11,773 GPS points) is less than 20 minutes (including the execution of the automated curve identification component at less than 30 seconds), the map-based QA/QC at less than 19 minutes), and the map-based result reporting at less than 5 seconds). More than 86% (80 out of 92 curves) of the curves were automatically extracted using the automated curve identification component, and the remaining 12 curves were also effectively inventoried using the map-based QA/QC component (e.g., add, delete, and editing curve PC and PT). The results show that the Smart-CIE tool provides an accurate and efficient means for transportation agencies to use when extracting curve information at the network level,

- which is essential to support transportation agencies' curve safety improvement programs (e.g., SafetyAnalyst, model inventory of roadway elements (MIRE), etc.).
- An extended component, HFST Start/End Points Identification, was also developed in the Smart-CIE tool using the extracted PC/PT and the Texas Transportation Institute (TTI) HFST installation guidelines to reliably and accurately determine the start and end points for HFST installation. The results show it is promising for transportation agencies to use the HFST Start/End Points Identification to consistently and accurately determine the starting and ending points of HFST installation, which results in the saving of the HFST costs and materials.
 - A case study using the HFST sites in a GDOT project (Project Identification Number, PI#0009993) on State Route 2 was conducted to demonstrate the capability of the extended component of HFST Start/End Points Identification to accurately and effectively determine the start and end points of the HFST installation. Results show the developed component is very promising for costing much less than the current manual procedure for determining the start and end points of the HFST installation. It has resulted in an estimated saving of \$600,000, which is, approximately, the cost of 25,000 sq. yd. of HFST material and other operations. This is much less than the manual determination of the start and end points of HFST installation using the same TTI HFST installation guidelines.
 - The research team developed detailed documentation and training materials, including a user manual, a tutorial video, and sample data.

The implementation of the research outcomes and recommendations for future research are presented below:

- It is recommended that training be conducted for implementation of the Smart-CIE, including one training session for GDOT, one training session for counties and cities through the GDOT Local Technical Assistance Program (LTAP), and one nationwide training session through the Federal Highway Administration (FHWA) Every Day Counts (EDC) program. To better disseminate the results, the training sessions may be provided by traditional class training or by webinar.
- It is recommended that GDOT conduct a further study to validate the curve information (e.g., curve radius, PC, and PT) extracted by different means, such as CARS (Curve Advisory Reporting Service) and Smart-CIE.
- It is recommended that GDOT implement a statewide curve inventory using the developed Smart-CIE to maximize the utilization of the GPS and GIS data already available to the Office of Transportation Data (OTD).
- It is also recommended that GDOT implement the extended component of HFST Start/End Points Identification HFST installation.
- It is recommended that a smartphone app be developed to further improve the extended component of HFST Start/End Points Identification for locating HFST start and end points in the field.

REFERENCES

1. Ai, C., and Tsai, Y. (2012). "Hybrid Active Contour-Incorporated Sign Detection Algorithm." *Journal of Computing in Civil Engineering*, 26(1), 28-36.
2. Ai, C., and Tsai, Y. (2014). "Critical Assessment of an Enhanced Traffic Sign Detection Method Using Mobile LiDAR and INS Technologies." *Journal of Transportation Engineering*, 141(5), 04014096.
3. Ai, C., and Tsai, Y. (2014). "Geometry Preserving Active Polygon-Incorporated Sign Detection Algorithm." *Journal of Computing in Civil Engineering*, doi:10.1061/(ASCE)CP.1943-5487.0000422, 04014092.
4. Ai, C., and Tsai, Y. (2015). "Automatic Horizontal Curve Identification and Measurement Method Using GPS Data." *Journal of Transportation Engineering*, 141(2), 04014078.
5. Camacho-Torregrosa, F. J., Pérez-Zuriaga, A. M., Campoy-Ungría, J. M., García, A., and Tarko, A. P. (2015). "Use of Heading Direction for Recreating the Horizontal Alignment of an Existing Road." *Computer-Aided Civil and Infrastructure Engineering*, 30(4), 282-299.
6. Dong, H., Easa, S., and Li, J. (2007). "Approximate Extraction of Spiralled Horizontal Curves from Satellite Imagery." *Journal of Surveying Engineering*, 133(1), 36-40.
7. Drakopoulos, A., and Örnek, E. (2000). "Use of vehicle-collected data to calculate existing roadway geometry." *Journal of Transportation Engineering*, 126(2), 154-160.
8. Easa, S., Dong, H., and Li, J. (2007). "Use of Satellite Imagery for Establishing Road Horizontal Alignments." *Journal of Surveying Engineering*, 133(1), 29-35.
9. ESRI (2010). "GIS for Highways." <<http://www.esri.com/industries/highways>>. (April 10, 2014).
10. FDOT (2010). "Geographic informationsystem Application Manager - Curvature Extension." <<http://www.dot.state.fl.us/planning/statistics/gis/>>. (April 10, 2014).
11. Findley, D. J., Zegeer, C. V., Sundstrom, C. A., Hummer, J. E., Rasdorf, W., and Fowler, T. J. (2012). "Finding and Measuring Horizontal Curves in a Large Highway Network: A GIS Approach." *American Public Works Association - Public Works Management and Policy*, 17(2), 189-211.

12. Imran, M., Hassan, Y., and Patterson, D. (2006). "GPS–GIS-Based Procedure for Tracking Vehicle Path on Horizontal Alignments." *Computer-Aided Civil and Infrastructure Engineering*, 21(5), 383-394.
13. Kasa, I. (1976). "A circle fitting procedure and its error analysis." *IEEE Transactions on Instrumentation and Measurement*, IM-25(1), 8-14.
14. Li, Z., Chitturi, M., Bill, A., and Noyce, D. (2012). "Automated Identification and Extraction of Horizontal Curve Information from Geographic Information System Roadway Maps." *Transportation Research Record: Journal of the Transportation Research Board*, 2291, 80-92.
15. Li, Z., Chitturi, M. V., Bill, A. R., Zheng, D., and Noyce, D. A. (2015). "Automated Extraction of Horizontal Curve Information for Low-Volume Roads." *Transportation Research Record: Journal of the Transportation Research Board*, 2472, 172-184.
16. Meek, D. S., and Walton, D. J. (1992). "Clothoid Spline Transition Spirals." *Mathematics of Computation*, 59, 117-133.
17. Pratt, V. (1987). "Direct least-squares fitting of algebraic surfaces." *SIGGRAPH Comput. Graph.*, 21(4), 145-152.
18. Rasdorf, W. J., Findley, D. J., Zegeer, C. V., Sundstrom, C. A., and Hummer, J. E. (2012). "Evaluation of GIS Applications for Horizontal Curve Data Collection." *Journal of Computing in Civil Engineering*, 26(2), 191-203.
19. Sarbaz, O., Gunnar, L., and Robert, T. (2012). "Using Naturalistic Field Operational Test Data to Identify Horizontal Curves."
20. Taubin, G. (1991). "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11), 1115-1138.
21. Tsai, Y., Kim, P., and Wang, Z. (2009). "Generalized Traffic Sign Detection Model for Developing a Sign Inventory." *Journal of Computing in Civil Engineering*, 23(5), 266-276.
22. Tsai, Y., Wu, J., Wang, Z., and Hu, Z. (2010). "Horizontal Roadway Curvature Computation Algorithm Using Vision Technology." *Computer-Aided Civil and Infrastructure Engineering*, 25(2), 78-88.

23. Xu, H., and Wei, D. (2016). "Improved Identification and Calculation of Horizontal Curves with Geographic Information System Road Layers." *Transportation Research Record: Journal of the Transportation Research Board*, 2595, 50-58.
24. Yun, D. G., Sung, J. G., and Kim, H. S. (2013). "Development and identification algorithm for horizontal alignment using attitude and positional data of vehicle." *KSCE Journal of Civil Engineering*, 17(4), 797-805.

APPENDIX A DETAILED CURVE FINDER ALGORITHM

Roadway Segmentation

The objective of roadway segmentation is to automatically partition the GPS trajectory data that represents the roadway into delineated segments. Each segment shares the same radius measurement. The iterative process is proposed to ensure that an optimized number of GPS points are grouped for each segment, while the circular fitting process is introduced to improve the robustness of the measurement to the disturbances in GPS points. In previous studies, automatic circular fitting methods have been proposed with accurate results (Kasa, 1976; Pratt, 1987; Taubin, 1991). However, it is not feasible to directly apply these methods for horizontal curve radius measurement without identifying the delineation of the roadway. These circular fitting methods only take a fixed number of points for circular fitting. Hence, no matter how robust the circular fitting method is, the fitting results are severely biased by the selected number of points.

In this study, an iterative circular fitting algorithm is proposed to the GPS points to obtain the initial circular fitting results. Figure A-1 shows the pseudo code for the iterative circular fitting and an example of how the iterative fitting converges to an optimal group of GPS points for each segment. Instead of selecting a fixed number of neighboring points for fitting, an incremental number of neighboring points are attempted until arriving at the least fitting error. The number resulting in the least fitting error will be associated with this group of GPS points (i.e. L_n). The fitting error is measured by the fitness of the actual GPS points compared to the approximated circle. As shown in Figure A-1, the error curve is monitored and recorded separately until the

global minimal value is arrived at among all the attempted iterations. Once the number of neighboring points is selected from the current group of GPS points (i.e. L_n), the next circular fitting will be started by skipping the current group of points. Hence, optimized numbers of GPS points are selected for each group of GPS points (i.e. segment) through iterations.

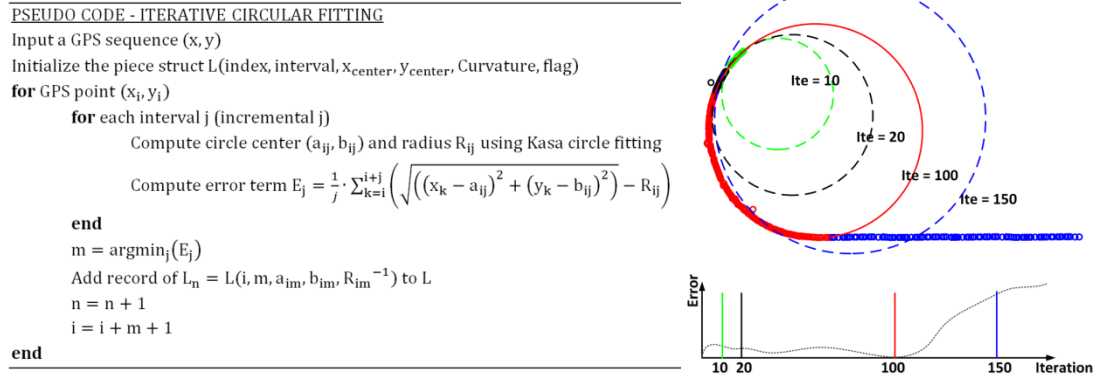


Figure A-1 Pseudo code and illustration for iterative circular fitting algorithm

In this study, the classic circular fitting method proposed by Kasa (1976) is selected for its computational efficiency and robustness to noise. The circular fitting method proposed by Kasa defines a minimization problem of the least square error between the sample points used for circular fitting and the approximated circle. The detailed description and the key equations in this method are derived in Appendix B. Figure A-2 shows an illustration of the resultant segments derived from the iterative circular fitting. The alternative colors used in the figure are to highlight the delineation. A single radius measurement is associated with all the GPS points within the same segment. Whether or not the delineated segment is curved or tangent will be determined in the subsequent steps.

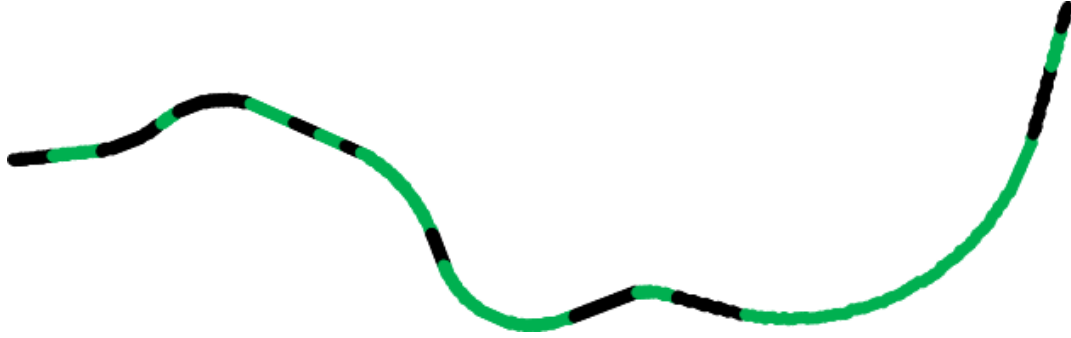


Figure A-2 Illustration of the resultant segments on a small road section

Curve Identification

The objective of curve identification is to automatically identify the horizontal curve and the corresponding curve types using the delineated segments derived from the previous step and based on the spatial distribution and adjacency of these segments.

Segment Type Identification: The types of the delineated segments are first determined using the center angle Δ . Figure A-3 shows the illustration of different center angles for curved and tangent segments. The center angle of the delineated segment can be computed using the fundamental arc length equation for circles, and the arc length can be estimated from the GPS acquisition parameter or directly obtained from the distance measurement instrument, as shown in the following equation:

$$\Delta_n = \frac{S_n}{R_n} = \frac{\sum_{i=1}^m v_i / f}{R_n}$$

where:

Δ_n – The center angle of the n^{th} segment in radian;

R_n – The radius computed by the iterative circular fitting for the n^{th} segment;

S_n – The arc length of the n^{th} segment;

m – The number of GPS points (i.e. interval) included in the n^{th} segment;

v_i – The instantaneous vehicle speed acquired by GPS at the i^{th} point;

f – The GPS acquisition frequency.

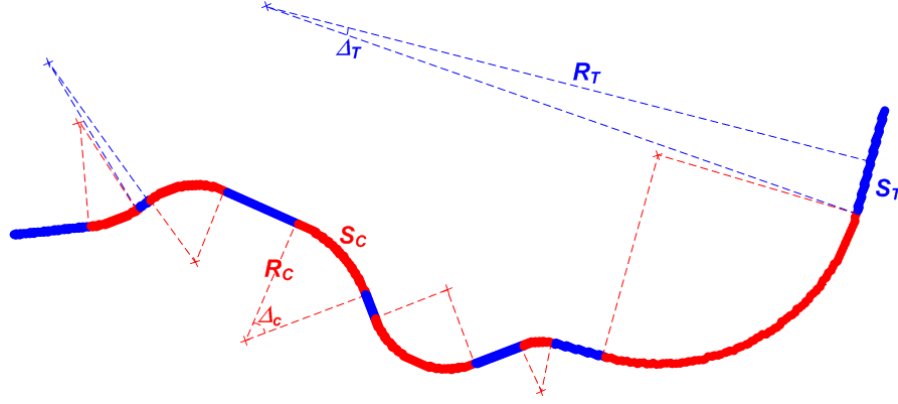


Figure A-3 Illustration of different center angles for curved and tangent segments

- For tangent segments, the error from the circular fitting will not be reduced by increasing the number of neighboring GPS points for circular fitting. Therefore, the group representing a tangent segment typically contains very few GPS points m and bears a large radius R_n , hence an extremely small center angle Δ_n .
- For curved segments, the error from the fitting will be reduced by increasing the number of GPS points until the selection of GPS points starts to include tangent sections. Therefore, the group representing a curved segment typically contains more GPS points (m) and results in a relatively small radius R_n , hence a relatively large center angle Δ_n .

An appropriate chosen Δ_n will impact the performance of the identifications of tangent segments and curved segments, hence the performance of the curve identification. Therefore, a parameter calibration process is needed to determine the optimal threshold of Δ_n . In this study, six roads were selected from the Savannah area in Georgia as the source of the calibration data and

covering different functional classes, e.g. interstate highway, state route and local road. The corresponding GPS trajectory data was collected and used to digitize the curves on these roads. The proposed iterative circular fitting was applied to segment the GPS trajectory data into delineated segments. A manual review was then conducted to determine whether the delineated segment is a tangent segment or a curved segment by correlating the segments to the manual digitization results. In total, 47 curve segments and 194 tangent segments were manually determined. Figure A-4 shows the distribution of the center angle Δ_n for these manual results. As shown in Figure A-4, the center angles derived from the tangent segments never exceed 5° , while the minimum center angle derived from the curved segments is 7.8° . Therefore, the threshold for the center angle Δ_n can be determined based on the distribution. In this study, the GPS acquisition frequency f is configured at 5Hz, and the threshold is set at 5° .

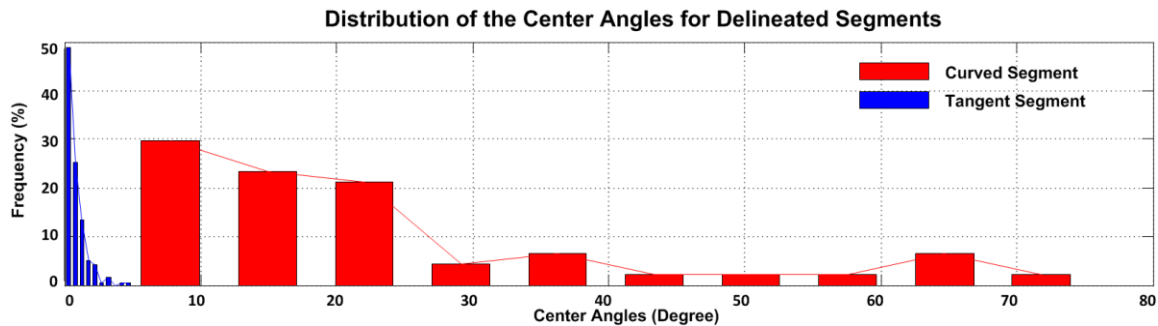


Figure A-4 Distribution of the center angles in the calibration dataset

Heading Validation. For each curved segment, the entering heading and the exit heading should be consistent with the deflection angle. Otherwise, the identified curved segment may potentially include more than one simple curve. This is particularly frequent when consecutive reverse or compound curves present the data. Therefore, a heading validation step is proposed to better determine whether a curved segment should be further segmented into multiple segments. If the difference between of the heading change (i.e., difference between the heading at the beginning

and ending of the curved segment) and the computed deflection angle is greater than 15° , the curve segment will be reprocessed using the iterative circular regression but with a smaller maximum iteration (i.e., half of the curve segment size). After this process, the tight consecutive reverse or compound curves will be correctly segmented. The output of the heading validation will be the original curved segment (if the criteria are not satisfied) or will be new curved segments (if the criteria are satisfied).

Curve Classification: The validated curved and tangent segments are then classified into different curve types according to their adjacency patterns. Figure A-5 shows the typical patterns for different curve types. Regardless of the curve type, there is always a back and forward tangent segment that enters and exits the center curved segment(s), respectively (i.e. shown as black in Figure A-5). Therefore, the pattern of the center curved segment(s) (i.e. shown as red and blue in Figure A-5) determines the curve types:

- Non-curve (Straight): consecutive tangent segments are presented;
- Simple curve: a single curved segment is presented;
- Compound curve: two curved segments with the same curve direction are presented;
- Reverse curve: two curved segments with opposite curve directions are presented;
- Spiral curve: three curved segments with the same curve direction and symmetric curvature measurements are presented.

Based on the patterns and the segment type results discussed above, the GPS trajectory data is re-examined. The center curved segment(s) is clustered with its adjacent tangent segments into a complete section representing different types of curves and straight lane.

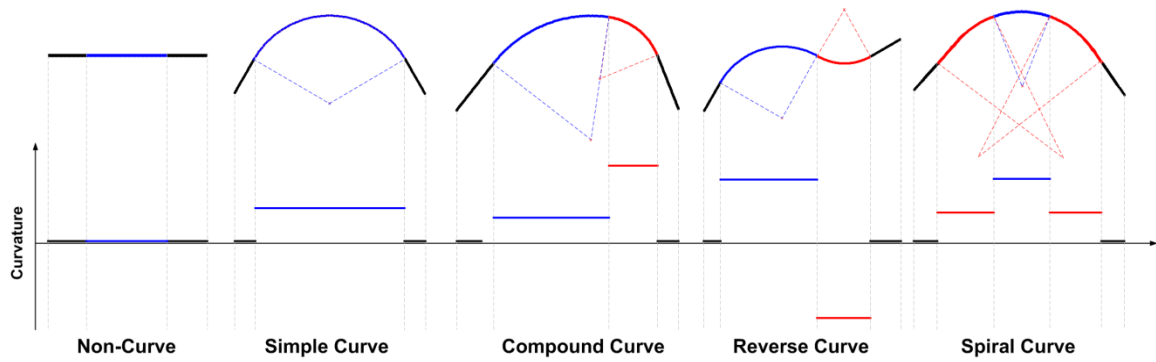


Figure A-5 Illustration of the segment adjacency patterns for different curve types

Spiral Curve Update

In previous steps, spiral curves are first identified as three consecutive curves, where the back and forward spirals are simplified using two simple curves with smaller curvatures. However, because the curvature of the spiral curve changes linearly with its curve length, circular fitting results in previous steps for these two spirals need to be updated. The Clothoid spline method (Meek and Walton, 1992) is used to approximate the spirals. From previous steps, the back spiral is connected between the ending point of the back tangent segment (TS) and the starting point of the center curve (SC), while the forward spiral is connected between the ending point of the center curve (CS) and the starting point of the forward tangent segment (ST). The detailed description and the key equations in the Clothoid spline method are presented in Appendix C.

As the equation for each identified curve has already been modeled in the step of iterative circular fitting and validated in the step of curve identification, the corresponding measurement of curvature (i.e. radius) is already computed and associated with each point of the GPS trajectory. By updating the curvature measurement at each point at spiral curves in this step, the horizontal curve identification and measurement processes are complete.

APPENDIX B KASA'S METHOD

Given a point set (x_i, y_i) , $i \in (1, 2 \dots N)$ for circular fitting, where d_i is the distance between each point and the center of the fitted circle (A, B) . Therefore:

$$d_i^2 = (x_i - A)^2 + (y_i - B)^2$$

The error between the edge of the fitted circle and the corresponding (x_i, y_i) is

$$\delta_i = d_i^2 - R^2 = (x_i - A)^2 + (y_i - B)^2 - R^2 = x_i^2 + y_i^2 + ax_i + by_i + c$$

Let

$$Q(a, b, c) = \sum \delta_i^2 = \sum (x_i^2 + y_i^2 + ax_i + by_i + c)^2$$

The objective is to derive a , b and c so that $Q(a, b, c)$ is minimized. As $Q(a, b, c)$ is none negative, it has a minimal value only when the following equation holds:

$$\begin{aligned} \frac{\partial Q(a, b, c)}{\partial a} &= \sum 2(x_i^2 + y_i^2 + ax_i + by_i + c) x_i = 0 \\ \frac{\partial Q(a, b, c)}{\partial b} &= \sum 2(x_i^2 + y_i^2 + ax_i + by_i + c) y_i = 0 \\ \frac{\partial Q(a, b, c)}{\partial c} &= \sum 2(x_i^2 + y_i^2 + ax_i + by_i + c) = 0 \end{aligned}$$

By solving the equations:

$$a = \frac{HD - EG}{CD - D^2}$$

$$b = \frac{HC - ED}{D^2 - GC}$$

$$c = \frac{\sum (x_i^2 + y_i^2) + a \sum x_i + b \sum y_i}{N}$$

where

$$\begin{aligned}C &= N \sum x_i^2 - \left(\sum x_i \right)^2 \\D &= N \sum x_i y_i - \left(\sum x_i \right) \left(\sum y_i \right) \\E &= N \sum x_i^3 + N \sum x_i y_i^2 - \left(\sum x_i^2 + y_i^2 \right) \left(\sum x_i \right) \\G &= N \sum y_i^2 - \left(\sum y_i \right)^2 \\H &= N \sum y_i^3 + N \sum x_i^2 y_i - \left(\sum x_i^2 + y_i^2 \right) \left(\sum y_i \right)\end{aligned}$$

So

$$A = -\frac{a}{2}$$

$$B = -\frac{b}{2}$$

$$R = \frac{1}{2} \sqrt{a^2 + b^2 - 4c}$$

APPENDIX C CLOTHOID SPLINE METHOD

Figure C-1 shows the setup of the Clothoid spline with a back spiral. The objective is to determine the back spiral function between TS and SC given the expected starting location of the spiral P and the starting (SC) and ending of the center curve (CS). The computation for the forward spiral follows the same steps without losing any generality.

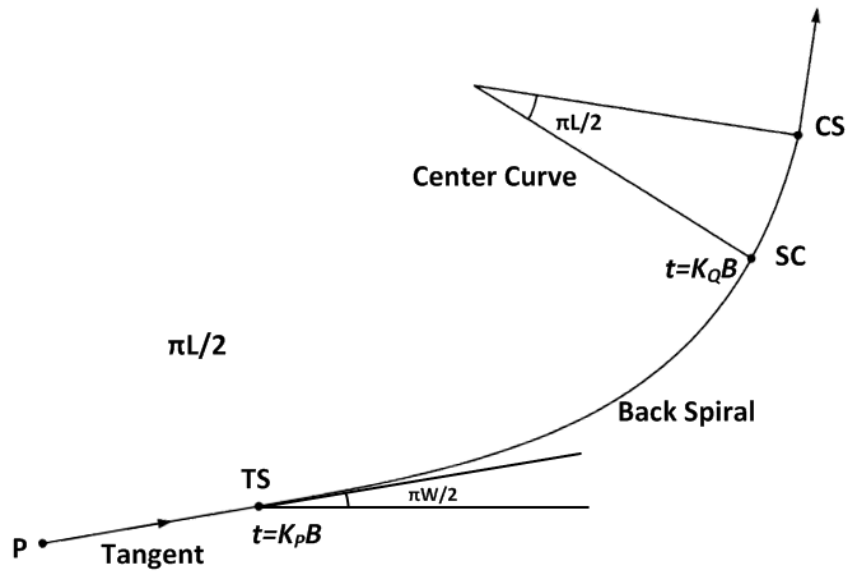


Figure C-1 Illustration of the Clothoid spline with a back spiral

Where:

K_P – Unit curvature entering the spiral, $K_P = 0$

K_Q – Unit curvature exiting the spiral, $K_Q = \text{curvature of the center curve}$

B – Scaling factor of the curvature

The spiral can be represented using Fresnel integral (Meek and Walton 1992): $\pi B \begin{pmatrix} C(t) \\ S(t) \end{pmatrix}$, where

$C(t) = \int_0^t \cos\left(\frac{\pi}{2}u^2\right) du$, $S(t) = \int_0^t \sin\left(\frac{\pi}{2}u^2\right) du$, and πB is the scaling factor. The center of the center curve can be computed based on the circular arc function:

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} + \frac{1}{K_Q} \begin{pmatrix} -\sin\left(\frac{\pi}{2}W\right) \\ \cos\left(\frac{\pi}{2}W\right) \end{pmatrix}$$

The center of the center curve can also be computed based on the spiral:

$$\begin{pmatrix} x + \frac{\pi}{K_Q} C_I(\sqrt{W-L}) \\ \frac{\pi}{K_Q} S_I(\sqrt{W-L}) + \frac{1}{K_Q} \end{pmatrix}$$

Therefore:

$$\begin{cases} x + \frac{\pi}{K_Q} C_I(\sqrt{W-L}) = Q_1 - \frac{1}{K_Q} \sin\left(\frac{\pi}{2}W\right) \\ \frac{\pi}{K_Q} S_I(\sqrt{W-L}) + \frac{1}{K_Q} = Q_2 + \frac{1}{K_Q} \cos\left(\frac{\pi}{2}W\right) \end{cases}$$

where:

$\frac{\pi}{2}L$ – The angle of center curve

$\frac{\pi}{2}W$ – The angle of the starting of the back spiral with respect to the horizontal

x – Distance between TS and P

Since the locations of P and SC are given, and the L is computed by the previous circular fitting, the spiral curve can be computed by solving the above equations for W and x. With the computed W and x, the total length of the spiral (S_0) can be computed. To effectively represent and record

the approximated spiral curve, slope factor k , together with the location of TS and SC, are recorded for each spiral curve in this paper.

$$k = \frac{S}{\frac{1}{R}} = \frac{S_0}{\frac{1}{R_0}} = S_0 \cdot K_Q \cdot B$$

where:

S – Length from a certain point to its initial position along the spiral

R – Radius at a certain point of the spiral curve

R_0 – Radius of the center curve (i.e. end of the spiral curve), equals $t=K_Q B$

APPENDIX D PROCEDURE FOR USING ARCOBJECTS ADD-IN APPLICATION

Import data

In this step, the user selects the roadway data for which curves need to be identified. This program accepts two types of inputs - CSV files and ArcGIS shapefiles files. The user clicks the “Import” button to load the roadway data into the program (Figure D-1).



Figure D-1 Import data

Upon clicking the "Import" button, a file browser opens, and the user selects the input file and a new form opens. In the form, in the case of user selected a CSV input, the user has to select the columns containing the "X Field" and the "Y Field" values from the drop-down menu. These columns represent the roadway alignment coordinates. For example, when the CSV file stores the roadway alignment in GPS coordinates, the longitude column of the CSV file is selected for the "X Field" and the latitude column is selected for the "Y Field," respectively, as illustrated in Figure D-2.

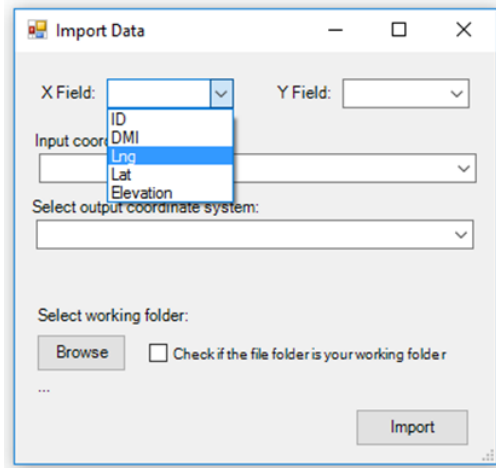


Figure D-2 An example of selecting 'X Field' and 'Y Field' for CSV inputs

For shapefile input, the users will not be prompted to provide any values. The predefined coordinate system of the shapefile will show up at the "Input coordinate system" drop-down box, as seen in Figure D-3.

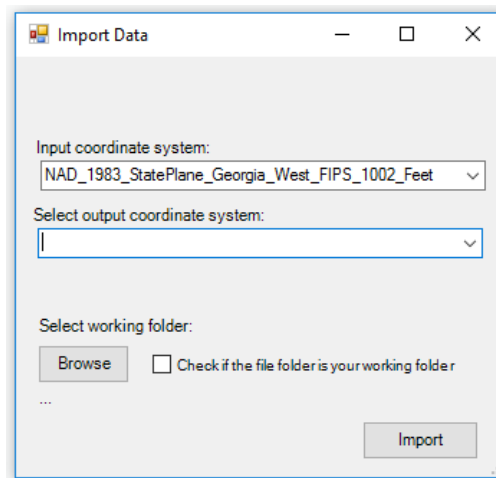


Figure D-3 The projection system automatically populates for shapefile input

Select output coordinate system

The output coordinate system is specified to convert the input coordinates into a user-selected, projected coordinate system. The curve-finder algorithm runs on 2-D coordinate values, and, consequently, the program needs to convert the 3-D geographic coordinates of the roadway alignment points into the 2-D coordinates.

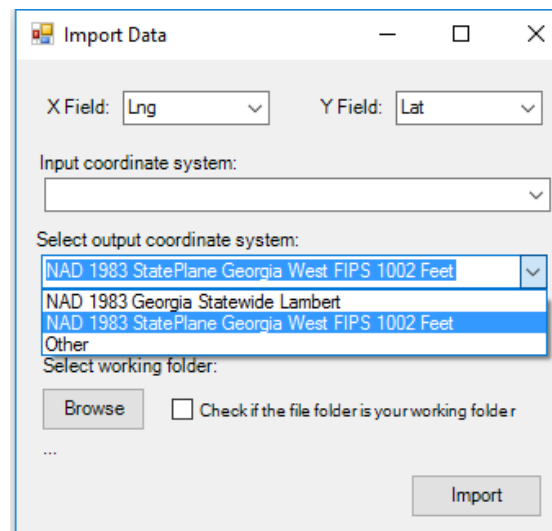


Figure D-4 An example of selecting the output coordinate system

For the CSV input, the user can select a coordinate system pre-loaded in the drop down menu, as shown in Figure D-4. For a shapefile input, if the input coordinate system is already in a projected coordinate system, users have to reselect the same coordinates in the dropdown. If the input coordinate system is undefined or in the geographic coordinate system, the user can select a coordinate system pre-loaded in the drop down menu.

Select working folder

The users are given an option to select the folder in their system in which all the outputs and intermediate files will be stored. The user can choose to select the same folder as the input file or choose a different folder by clicking the "Browse" button.

Import

The Import Data form is filled, as shown in Figure D-5, for the two input types. The user must finally click the "Import" button to convert the input coordinates into 2-D coordinates, and the program allows the user to visualize the roadway segment on the map. The rest of the program will use the converted coordinates to run all the processes.

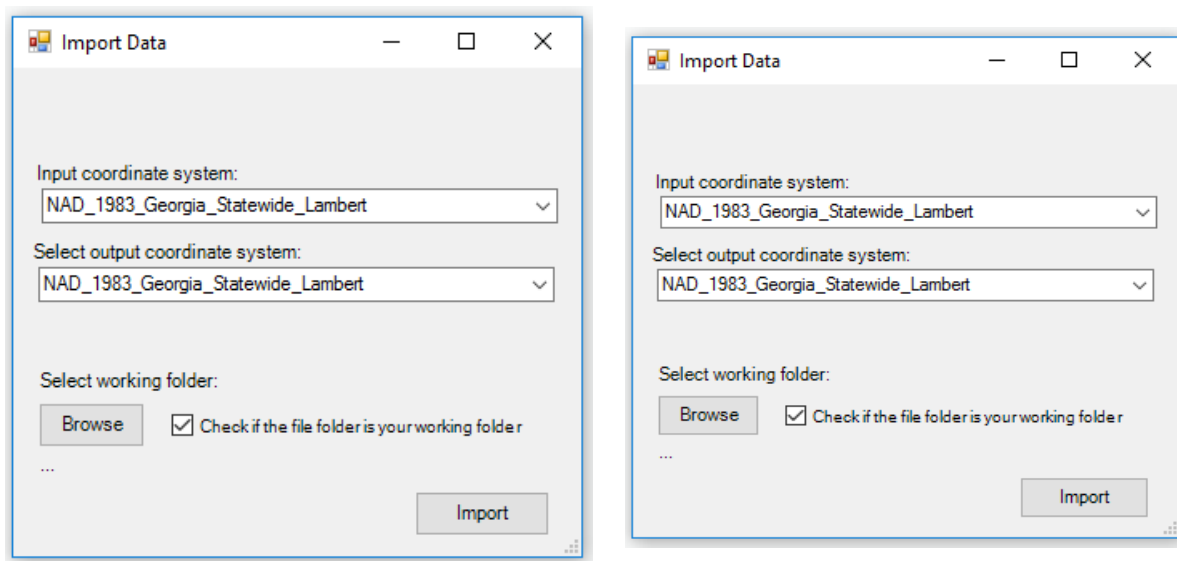


Figure D-5 An example of filling up Import Data forms after selecting the working

Step 2: Identify Curve

The user then clicks the "Identify Curves" button to run the curve-finder algorithm on the roadway segment and display the curves identified on the roadway on the map (Figure D-6 and Figure D-7). The user can also see the details of the curve, such as radius, deviation angle, coordinates of endpoints, etc. in the user interface. The user can also choose the category of parameters from the "Curve finder category" dropdown box before clicking the "Identify

Curves" button. The default category is "General Curves." This category filters out curves whose radii are more than 10,000 feet and whose deviation angles are less than 1.7 degrees.

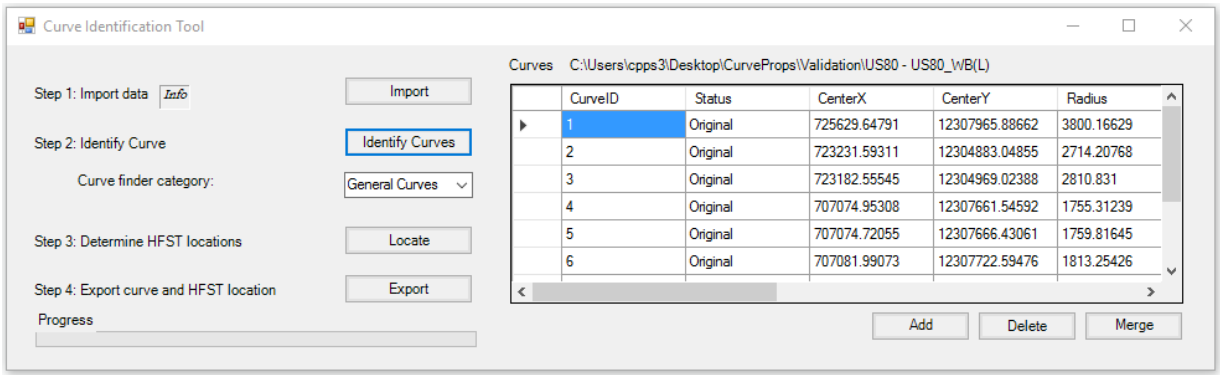


Figure D-6: An example of curves identified by the tool

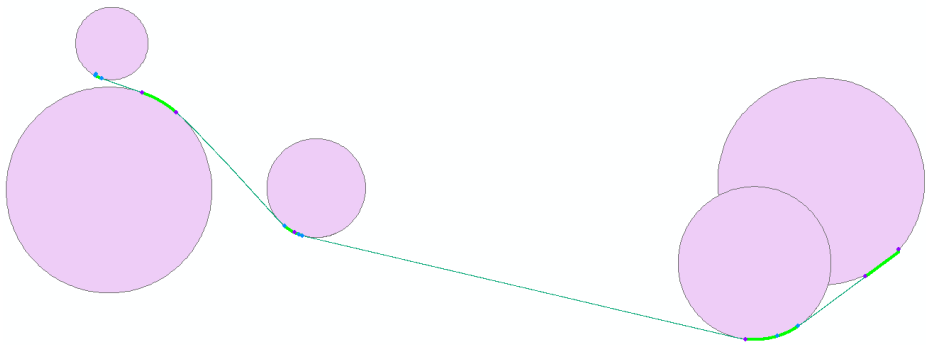


Figure D-7 A visualization of the curves identified by the tool

Step 3: Determine HFST locations

Once the curves are identified, the next step is to determine the HFST endpoints, which is achieved in this step. The user selects the curves for which HFST endpoints need to be located and clicks on the "Locate" button (Figure D-8) that opens a new interface (Figure D-9).

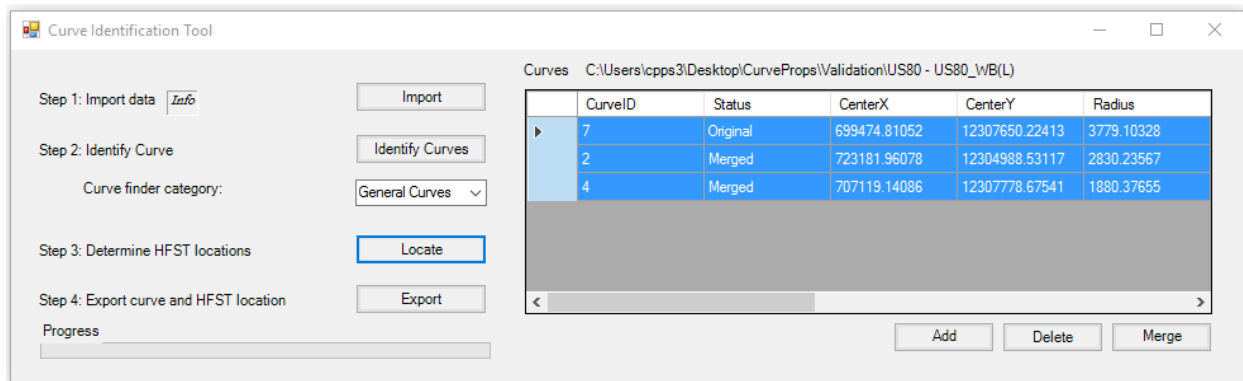


Figure D-8 An example of selecting curves to locate HFST endpoints

In the new interface, the user must enter the posted speed and the advisory speed assigned for each curve and click “Determine HFST Endpoints,” as shown in Figure D-9. HFST endpoints are determined using the Texas Transportation Institute recommendations based on two speeds. The results are displayed in the user interface table, which includes the upstream HFST length from the point of the curve in either direction, coordinates of the endpoints, and the curve details (Figure D-10). The locations are also highlighted on the map (Figure D-11).

StartingInd	PointsONCurve	Posted Speed	Advisory Speed
894	92	55	45
81	247	55	35
697	182		
*			

Figure D-9 User inputs posted and advisory speed for each curve

HFST End Points - TTI Recommendation

Enter posted and advisory speed limits of the curves in the table

Curve	Posted Speed	Advisory Speed	UpstreamHFSTLength	HFSTStartX
	55	45	108	706688.8333
	55	35	194	724933.3853
▶	55	45	108	702041.7214
*				

Scroll horizontally to view HFST end points

Determine HFST End Points Refresh

Figure D-10 An example showing the populated HFST table

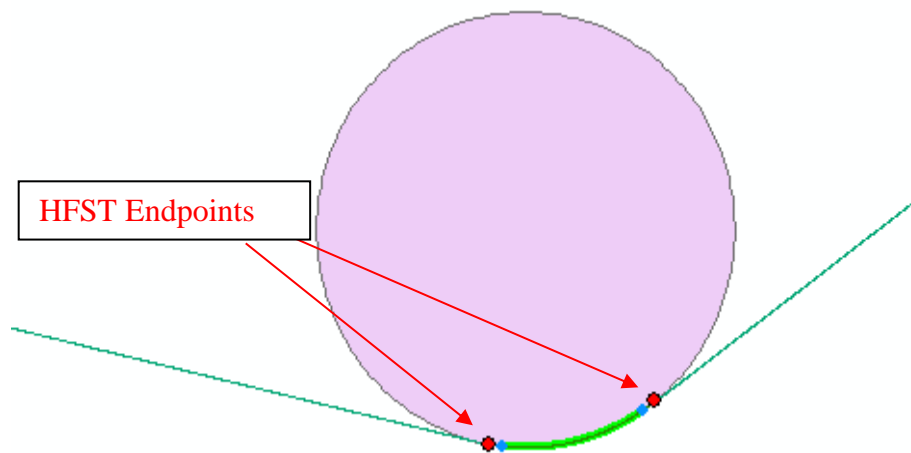


Figure D-11 A visualization of HFST endpoints

Step 4: Export curve and HFST location

The details in the curve table and HFST location information table are exported into a CSV format in the user selected folder by clicking the "Export" button (Figure D-12).

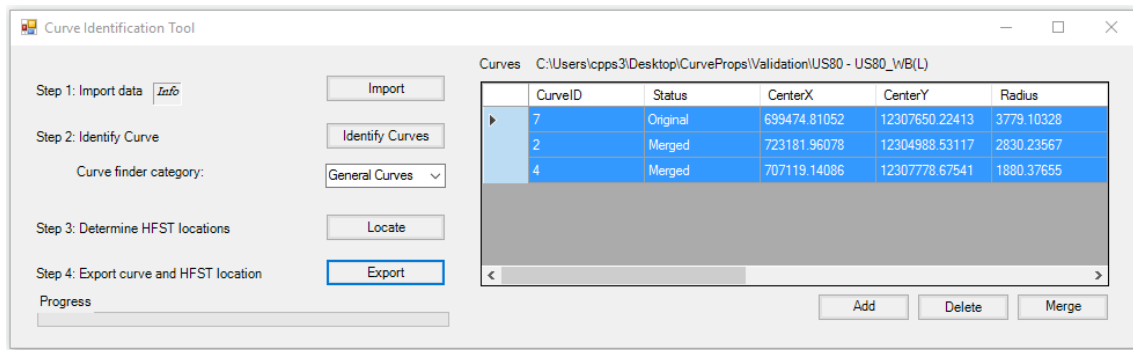


Figure D-12 Click 'Export' to save the results in local system