ArcGIS Horizontal Curve Recognition and Fitting Tool

Douglas Wright
In Brief:

• Developed by AECOM in 2014
• Python tool in ArcGIS for horizontal curve detection and fitting of digitized curves in transportation safety analyses
• Generates an algebraic solution with negligible inbuilt bias
• Detects curves and generates best-fit parameters in minimal time for large road vector datasets
The Challenge

The Kansas Department Of Transportation (KDOT) required a tool to generate best-fits for horizontal curves for their freeway system that must:

- Run in ArcGIS
- Detect the start and end locations of all curves within digitized road polyline feature classes
- Provide output feature classes comprising detected curves and the best-fit curve with associated statistics
Background

• Horizontal curves are circular curves, with a radius and center of curvature

• Road safety is best-served by circular curves because this allows the driver to negotiate the curve without having to change the tilt of the wheel while passing through

• Detection of locations where curves depart from circularity is necessary for safety evaluations, and remediation measures such as
  - Stopping distance
  - Passing distance
For example

- In this case the road does not require a driver to change steering direction in the curve and the force exerted on the vehicle does not change.
In contrast,

- Curves of changing radius are intrinsically unsafe, such as this notorious case:
Literature Review

• No other similar applications were found in literature review as of 2014

• Existing research:
  • Zhixia, et all.
  • Ali Al-Sharadqah and Nikolai Chernov, 2009

• We incorporated the most promising techniques into our code
• Regression Techniques using Iteration
  - Also called “Geometric”
  - Minimizes the function:

\[
F(a, b, R) = \sum d_i^2,
\]

where \( d_i \) stands for the distance from \((x_i, y_i)\) to the circle, i.e.

\[
d_i = r_i - R, \quad r_i = \sqrt{(x_i - a)^2 + (y_i - b)^2},
\]

where \((a, b)\) denotes the center, and \(R\) the radius of the circle.
Solutions
Programming Approaches using Regression techniques

- **Advantages**
  - Minimal error

- **Disadvantages**
  - Computationally intensive
  - May take many iterations to converge to a solution
  - Longer processing time
  - Some cases may never converge to a solution
Solutions

Programming Approaches using Algebraic Methods

• Algebraic
  - Non-iterative
  - Achieves result by solving for the minima in a system of linear equations
  - Advantages
    - Fast processing time
  - Disadvantages
    - Larger error than geometric methods
• **Algebraic “Hyperfit” Algorithm**
  - Developed by Ali Al-Sharadqah and Nikolai Chernov, Department of Mathematics, University of Alabama at Birmingham, 2009
  - Has negligible inbuilt bias and error
  - Outperforms geometric fits in accuracy
  - Performs extremely fast
  - Python code publically available for commercial or private use
Program Outline

In summary:

- Developed as a Toolbox employing Python code (v 2.7) and Numpy
- Filters input data with topological rules
- Requires user to specify input parameters
- Reduces input route lines into vertices for detection of curve starts and ends, and for use in Hyperfit algorithm
- Creates features classes for
  - Segments of input detected as curves
  - Generates output curve from Hyperfit solution using the associated Radius and Center of Curvature.
  - Generates R-squared ratings for curves
## Input Parameters

![KDOT Horizontal Curve Tool](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Roads</td>
<td></td>
</tr>
<tr>
<td>Workspace</td>
<td></td>
</tr>
<tr>
<td>Name Fields: Provide a field containing the road names</td>
<td></td>
</tr>
<tr>
<td>Perform Topology Check (optional)</td>
<td></td>
</tr>
<tr>
<td>Simplify Line Tolerance</td>
<td>0.00328084 Feet</td>
</tr>
<tr>
<td>Bearing Change Threshold (degrees)</td>
<td>1</td>
</tr>
<tr>
<td>Maximum Distance</td>
<td>1040.33 Feet</td>
</tr>
<tr>
<td>Output Observed Curve Lines</td>
<td></td>
</tr>
<tr>
<td>Output Perfect Fitted Curves</td>
<td></td>
</tr>
<tr>
<td>Angle Resolution for Output Perfect Fitted Curves</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Considerations for Input

Recognizes start and end of curve by changes in bearing within input tolerance or if the bearing reverses direction (e.g., clockwise into anti-clockwise)
Considerations for Input

However, there may be data defects, where digitization errors create a reversal in direction that will prematurely end curve:
Considerations for Input
Data defects, where multipart route lines with the same name have opposing directions
Considerations for Input

**Maximum distance:** This is an input parameter of a maximum distance between two points in a curve to ensure the curve fitting process calculates a realistic result, and can be chosen based on a consideration of the straight-line distance required to stabilize a vehicle at the end of a horizontal curve when driving along the curve at full speed.
Data Treatment

• Removes divots by applying the “Simplify Line” tool with a specified input tolerance
• As an option, applies a topology rule to detect the presence of multipart lines and remove them from analysis
• Generates unique identifiers for each line in case there are duplicate road names
Considerations for Output

Exports each segment of recognized curves to a feature class
Considerations for Output

Generates a feature class with all solutions including output parameters and statistics, including R-squared ratings
Considerations for Output

Generates a feature class with all solutions including output parameters and statistics, including R-squared ratings.
Conclusions

- Alternative technique using unique algebraic method superior to regression techniques
- Expands the range of GIS applications for transportation safety
- Effective, fast tool that runs totally in GIS for combined curve recognition and fitting
  - Analyzed the entire Kansas state highway system in 15 minutes
Questions and Answers