SAND2018-12723

CURVATURE BASED ANALYSIS TO IDENTIFY AND CATEGORIZE TRAJECTORY SUBSEGMENTS

PAUL T. SCHRUM, JR.*, MARK D. RINTOUL†, AND BENJAMIN D. NEWTON‡

Abstract. Since the attacks carried out against the United States on September 11, 2001, which involved the commandeering of commercial aircraft, interest has increased in performing trajectory analysis of vehicle types not constrained by roadways or railways, i.e., aircraft and watercraft. Anomalous trajectories need to be automatically identified along with other trajectories of interest to flag them for further investigation. There is also interest in analyzing trajectories without a focus on anomaly detection. Various approaches to analyzing these trajectories have been undertaken with useful results to date. In this research, we seek to augment trajectory analysis by carrying out analysis of the trajectory curvature along with other parameters, including distance and total deflection (change in direction). At each point triplet in the ordered sequence of points, these parameters are computed. Adjacent point triplets with similar values are grouped together to form a higher level of semantic categorization. These categorizations are then analyzed to form a yet higher level of categorization which has more specific semantic meaning. This top level of categorization is then summarized for all trajectories under study, allowing for fast identification of trajectories with various semantic characteristics.

1. Introduction. Whenever an object moves, its path is called a trajectory. For vehicles, these trajectories may be recorded over time by occasionally sampling the location in a certain coordinate reference frame. The resulting sequence of time-stamped position samples constitutes the "trajectories" we study in this research.

Of these vehicles, aircraft and water-borne vehicles are generally not constrained by predetermined pathways, as is the case for road-borne and rail-borne vehicles. Thus aircraft and watercraft have a large number of possible variations in path from one point to another. Furthermore, aircraft and watercraft have been used as transportation for threat agents specifically because of this greater freedom of course coverage, and in some cases the vehicles themselves have been used as the destructive weapon. It would be advantageous if unusual trajectories could be identified by analyzing their geometry as described by the sequence of sampled points. Because of these and other motivations, several researchers have undertaken to analyze and characterize trajectories from the recorded point sequences.

One general approach to trajectory analysis involves computation of the normalized parameter space or normalized feature space. This approach is used by [1], and [7]. [1] performs an image parsing technique, Recursive Multi-frequency Segmentation, cross applied to the speed parameter of bird flights to detect speed-based segments. [7] computes whole-trajectory aggregate values including total distance, and convex hull aspect ratio.

The method we have explored in this research takes a sequential point-by-point approach, looking at the geometry of the trajectory at each point and its nearest previous and next neighbors along the ordered sequence. This approach yields information at a detailed level which is then aggregated into groups of similar properties until the entire trajectory has been partitioned into meaningful subsegments. The 'meaningful subsegments' are referred to as Level 1 Categorizations, which are the primary result yielded by the method. The Level 1 Categorizations are reduced to data formats which are well suited to established data mining techniques and may be used for various purposes.

We present the method, which we call "The Curvature Parsing Method" (CPM), demonstrate its capability, and assess its strengths and weaknesses.

Although the method is intended to be generalizable to maritime trajectories, only aviation data was used in the development of these algorithms, so further discussion focuses mainly on aviation trajectories.

*North Carolina State University, ptschrum@ncsu.edu
†Sandia National Laboratories, mdrinto@sandia.gov
‡Sandia National Laboratories, bdnewto@sandia.gov

2. Methods. The core of the Curvature Parsing Method is shown below. The subsequent discussion on the method follows this pseudocode as an outline.

1. Load batch of trajectory data into memory.
2. For each trajectory:
   (a) Compute local point geometry, including point triple curvature.
   (b) Perform Level 3 Processing: Categorize Each Individual Point.
   (c) Level 2 Processing: Group adjacent similar Level 3 Nodes into subsequences and categorize.
   (d) Level 1 Processing: Aggregate Level 2 Nodes into subsequences based on sequence patterns and categorize.
   (e) Generate intermediate data outputs as requested for each trajectory.
3. Generate report summarizing results for each Level 1 Node for the batch.

The items in this listing serve as an outline for the Methods section.

Optional data outputs are
1. a kml file for visualization using Google Earth^{TM} (.kml) [1] ,
2. a plot of the parse tree graph (.png),
3. a detailed arc data report including derived geometry parameters (.csv),
4. a statistical summary report of Level 1 categorizations (.csv),
5. a Level 1 hash bin report grouping all trajectories by a hash string of the Level 1 categorization. (.csv)

The details of these output types are provided in Section 2.3.4. Figures 2.1 and 2.2 together illustrate two of these outputs for a single trajectory. Figure 2.1 shows a kml plot of a flight which begins in the cruise phase over Lake Erie and ends with a landing at Ottawa.

The parse tree graph of Figure 2.2 illustrates important features of the process, detailed below, including the way in which points constitute the leaves of the parse tree, the entire trajectory is the root of the tree, and Level 1 and 2 categorizations are between the root and the leaves.

2.1. Trajectory Data Source. Trajectory data used in the development of this project originates with Aircraft Situation Display to Industry (ASDI), a data feed of aircraft trajectories in United States and Canadian airspace provided by the Federal Aviation Administration (FAA). ASDI data includes "aircraft scheduling, routing, and positional information." [11] Our purposes only require use of the temporal positional information and the aircraft identity to distinguish individual trajectories.

The FAA obtains ASDI data from radar stations across North America, which gather positional information as well as information squawked by the aircraft such as identity and altitude. The data has not been optimized to eliminate anomalies. Specifically, the data sometimes contains points which are the result of two radars reporting the position of the same aircraft at slightly different positions and different time stamps, resulting in a zigzag pattern in occasional portions of the trajectory. We refer to this kind of data anomaly as radar jitter. Further, there are rare data anomalies in which two sequentially adjacent points have identical coordinates and time stamps. We refer to these as double-stamps. Both kinds of anomalies cause problems for the algorithm. Thus those points are not removed from the dataset, but the categorizer ignores them by continuing the previous categorization across the anomalies.

[1] It is also possible to output all kml files into subdirectories named for Level 1 hashes.

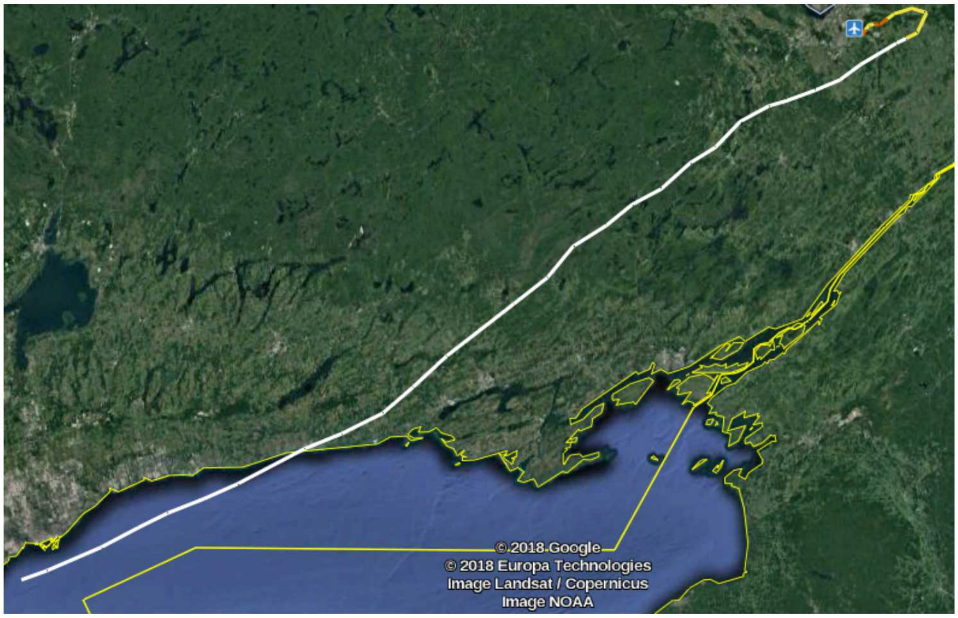

Fig. 2.1. *Partial Trajectory of Flight A3A458 in which sampling begins mid-flight over Lake Erie and ends with landing in Ottawa. White segments are straight; Yellow segments are left turns; Orange segments are right turns.*

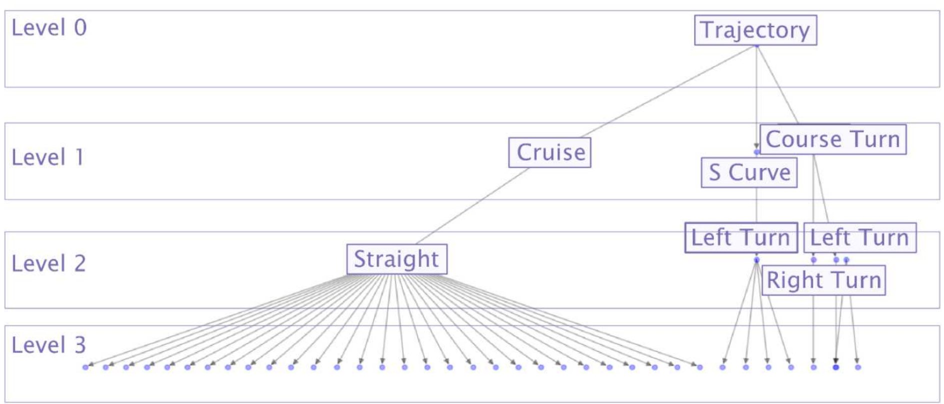

Fig. 2.2. *The Parse Tree Graph of Flight A3A458, depicted in Fig.2.1.*

Rintoul et al., in working on Sandia National Lab's PANTHER project, call for "a more thorough analysis of the information content in the different features" to be carried out, and an "examination of more efficient ways to break up the trajectories into segments to find smaller features". ([8], p. 72) The current research is part of that follow on work. The software developed for the research being presented here is an exploratory extension of Tracktable. The PANTHER report states, "Tracktable is an open source library which contains a core set of functionality for ingesting, processing, plotting, and analyzing trajectories." ([8], p. 39) The software implementation of the current research depends on Tracktable's Application Programming Interface (API) for data access and some computational geometry functionality. The desired outcome is a way to identify meaningful trajectory subsegments in partial support of a subset of PANTHER's stated goals:

1. finding trajectories that exhibit a behavior of interest without regard to translation,

rotation or scale,
2. dividing trajectories into specific clusters,
3. finding trajectories that are outliers with respect to a given set of trajectories, and
4. performing analysis prelimanry to classifing trajectories using unsupervised learning techniques.

([8], p 18)

In some cases the categorizations performed in the current research depend on trajectory subsegment absolute length, which makes the analysis no longer invariant under scale transforms.

2.2. Compute Local Point Geometry. The basis of this algorithm is the notion that any three points define a portion of a circle. Figure 2.3 shows how three coplanar points uniquely define an arc segment regardless of the offset of the middle point.

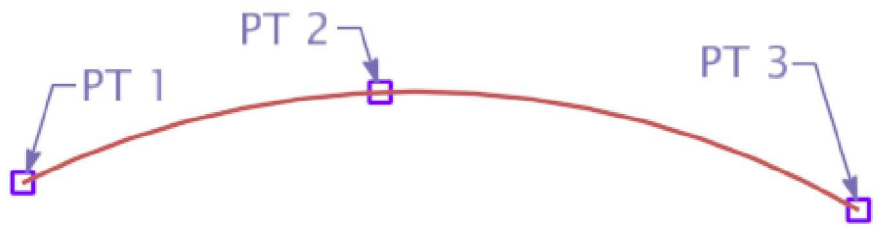

FIG. 2.3. *Three points defining a portion of a circle. The interior solution is shown. Exterior solutions, the arcs going the long way around the are not used in the present algorithm.*

Although the altitude input parameter is available for most trajectories, this algorithm does not use it, so all points are considered coplanar for the purpose of analysis. We are also developing an altitude based analysis in parallel with this research. At some point these two types of analyses may be merged for greater insight into aviation trajectories.

For each sequential point triplet of the given trajectory, the arc segment defined by point subsequence n-1, n, and n+1 is computed and the resulting computed information (hereafter referred to as 'curve data') is associated with point n. Hereafter, the midpoint of the point triplet, n, is referred to as the keypoint because in the data model, the arc segment parameters for arc n are stored with point n The first and last points of the sequence cannot have an associated arc, having only a single neighbor.

The terms 'trajectory' and 'alignment' are related but different. In the following discussion, 'trajectory' refers to the historical path of a physical object represented by a point moving through time; 'alignment' refers to a static, mathematically defined path independent of time. To clarify this distinction, roads are built along alignments, but cars trace out trajectories.

In fact, the disciplines of railway and roadway design, along with surveying, also concern themselves with determining the mathematical definition of smooth alignments from sampled data points. For this reason, some of the theory from those disciplines may be adapted for our purposes.

Concepts, symbols, and terminology are taken from [3], which is used extensively by surveyors and civil engineers for alignment definitions. Figure 2.4, adapted from [3] Figure 9 and accompanying discussion (pp 64-67), depicts the engineering symbology and definitions for the arc segment used to fillet two straight line segments. All geometry is flattened to eliminate the Z dimension. Further, the notion of Length in the following discussion only considers the 2D length of a path in the plane, and does not include any Z component. The parameters shown here are the curve Radius, R, and the curve deflection, Δ. PC is Point of Curvature, shared by the incoming tangent and the curve, PT, Point of Tangency, shared by the curve and the outgoing tangent, PI, the Point of Intersection of the two tangents without the curve, and CC, the Curve Center.

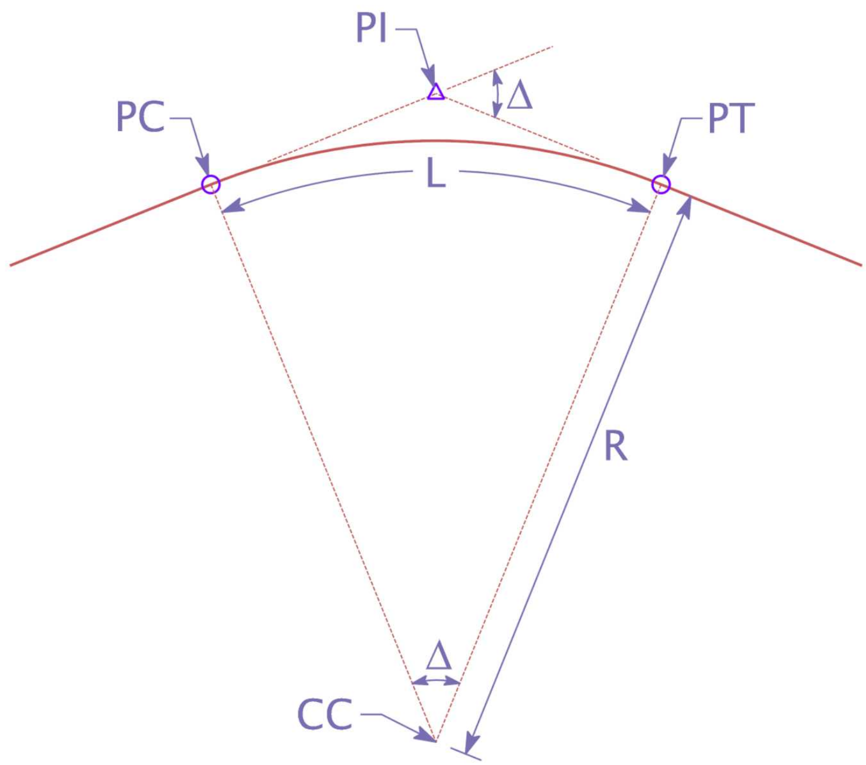

Fig. 2.4. *Basic arc segment elements using Civil Engineering terminology. CC is the Curve Center. PI is the Point of Intersection of the tangential line segments at the begin and end of the curve. PC is "Point of Curvature", which is the point where the alignment transitions from the line segment to the arc segment. PT is "Point of Tangency", which is the point where the alignment transitions from the arc segment to a different line segment. L is the length along the arc segment. Δ, Deflection, is the total change of heading of the arc segment from the PC to the PT.*

Curvature, also referred to as Degree of Curve (Dc), may be visualized as the degrees of heading change over one unit length along the trajectory or alignment in which the length of the arc changing by one degree is the same unit length for radius. Roadway design engineers in the United States use a length basis of 100 feet for Degree of Curve. For our research we work with a 1 mile length Degree of Curve basis. Figure 2.4 illustrates this concept. For the case where total curve length, L, equals 1 basis unit, then the total deflection, Δ, equals the curvature of the curve in degrees of change of heading per basis unit of length.

Several observations should be noted regarding Figure 2.4. The elements and equations for the engineering definition of an arc fillet are constrained by the requirement that all elements be continuous in heading. Curvature is the first derivative of heading. When a civil engineering alignment is used without Euler Spiral Segments (as depicted in Figure 2.4), there is a curvature discontinuity at the PC and PT, but heading is continuous. Following such an alignment precisely is physically impossible for real world vehicles due to conservation of momentum, so there are always easement curves at the beginning and end of a filleting curve which serve to ease from zero curvature to non-zero curvature and back. Civil engineers use Euler Spiral Segments to accomplish this easement, but easement curves are not being considered for this trajectory analysis.

Since curvature is the first derivative of heading, plots of heading versus length may conveniently be collocated on the same graph as curvature versus length. Figure 2.5, below, depicts such a simultaneous plot.

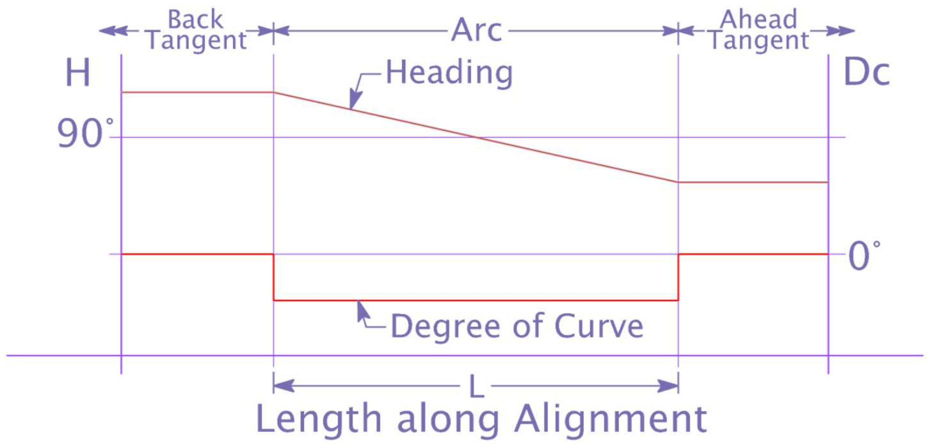

Fig. 2.5. *Heading versus Length Along (top line) and Degree of Curve versus Length Along Alignment (bottom line). This collocated plot illustrates the first derivative relationship between Heading and Curvature. This plot matches the arc segment shown in Figure 2.4.*

2.2.1. Limitations of Sampled Data. The plot in Figure 2.5, heading versus length combined with curvature versus length, is idealized for a proscribed engineering alignment such as the one shown in Figure 2.4. Sample-based trajectories have limitations due to sample noise and sampling rate. Regarding sample noise, if an aircraft is following a perfectly straight geodesic (also known as a Great Circle Route, the spherical equivalent of a straight line), noise will introduce errors into the reported position at each sample, leading to each point being slightly off the actual trajectory, and the curvature, which would be zero for a straight line, will have values near zero on both the positive side and the negative side, averaging to near zero over multiple sequential samples.

Our dataset consists of points sampled about every 60 seconds in most cases. Since the sampling rate is not continuous, certain turning events may be skipped over or under sampled, resulting in ambiguous or unresolvable features. This phenomenon is conceptually related to the Nyquist-Shannon Sampling Theorem [10].

The sampling rate is determined in the time domain, but the comparison basis for sampling rate is in the length domain. So a slower aircraft has a shorter physical distance between samples (more samples per hundred miles) than a fast moving one, and this has a higher sample resolution even though both have the same number of samples per minute.

A related problem is partial aliasing of point triplets at arc boundaries. Specifically, if a given point is near the beginning or end of a curve, one of its neighbor points lies on the adjacent tangent, and one lies on the curve. The result is that the key point (the middle point of the triplet) shows a transitioning curvature value even though that point is on the tangent of the alignment. This is illustrated in Figure 2.6

Figure 2.6 shows a contrived example of of sample points along a perfect alignment with no sample noise. The distance between samples is 6.0 miles. The sample arc associated with point 2 is highlighted in blue. Sample arcs 2, 3, 6, and 7 overlap the end points of the true curve, so their curve values are aliased between an arc and a tangent. Sample arcs 4 and 5 (not shown as arcs, but as points) fall completely on the true arc, so their curvatures will be the same as the true arc.

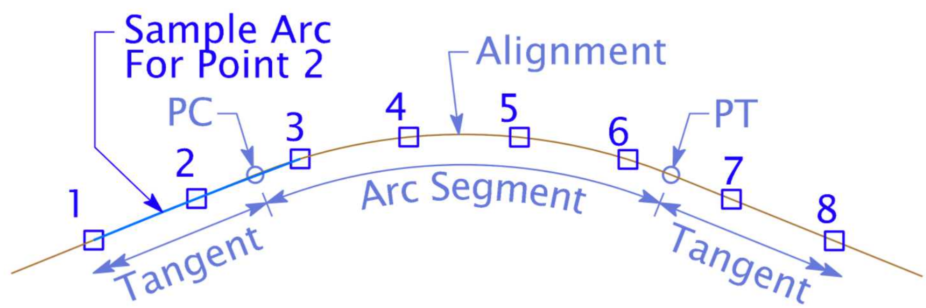

FIG. 2.6. *Sample points along an alignment showing aliasing where some point triplets span an end of the arc segment*

Arc Segment	R	Dc	Δ_{Sample}
1-2-3	32.57 mi	1.76°	2.11°
2-3-4	3.59 mi	15.97°	19.17°
4-5-6	3.00 mi	19.06°	22.89°
True Curve	3.00 mi	19.06°	43.98°

TABLE 2.1

Transition Values for Curvature (Dc) for point triplets spanning the beginning of a curve. R is sample curve radius. Δ_{Sample} *is sample curve deflection.*

Table 2.1 illustrates the same effect numerically.

In addition to end-aliasing, certain curves of short length may be skipped by sample points altogether, while the triple-point curvature data yields curvature and Δ values which differ considerably from the actual value. In the extreme case, the curve length is so short compared to the sample rate that sample points fall before and after the arc segment, but none fall directly on it, so aliasing occurs at all points involved with the curve.

Another caveat of basing the analysis on curve segments of sampled data points is that neither chord deflection δ_{Chord} or total arc deflection Δ_{Sample} represent the true change of heading from Point 1 to Point 3 when curves are under sampled. In the categorizations process, where change of heading is used for categorization criteria, it is based on δ_{Chord} because the estimation error is much less.

Figure 2.7 illustrates the definitions of these terms. The alignment being sampled is the same alignment as shown in 2.4 and 2.5, although the sample points are sparser. Δ_{Sample} is the deflection of the arc generated from the point triplet, 1-2-3, shown in blue. δ_{Chord} is the deflection of the straight line segments 1-2 and 2-3. This figure also demonstrates the effect of the sampled arc curvature and radius being substantially different from the true curve when only one sample point falls on the arc. We believe this effect impacts the subsequent categorization processing, but the details have not been confirmed and should be investigated further.

2.3. Categorization Parse Tree. After the algorithm computes local point geometry for all interior points, the points are categorized and grouped by categorization. Categorization is carried out in a multi-level process. The levels are labeled on Figure 2.8, which is a recapitulation of Figure 2.2. In Figure 2.8, Level 0 is at the root of the parse tree, the whole trajectory, and deeper levels progressing downward on the parse tree graph, down to Level 3, which has one categorization for each interior point (points 1 through $n-1$).

2.3.1. Level 3 Processing. The leaves of the parse tree are individual key points. Categorization starts at the bottom of the graph and works its way up. Level 3 processing

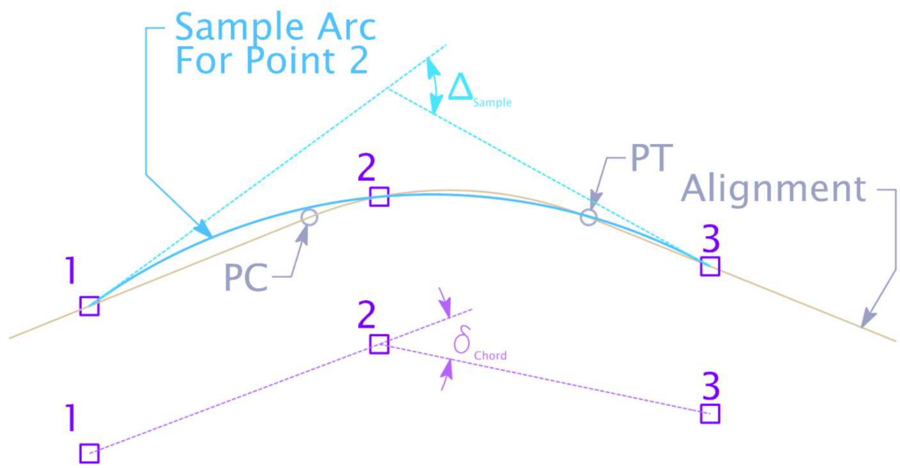

FIG. 2.7. *Different types of deflection from different computations. Note the point triplet 1-2-3 is repeated in the figure to improve readability.*

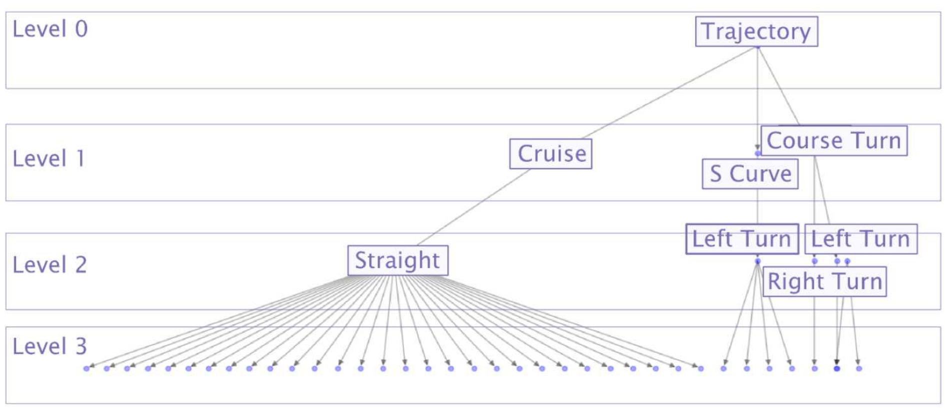

FIG. 2.8. *The Parse Tree Graph of Flight A3A458, depicted in Fig.2.1.*

is carried out first, which is based on the curve data computation of point triplets described above. There is one Level 3 node for each interior point of the trajectory.

Level 3 categorizations are grouped into three aspects: curvature, leg length, and point deflection (δ_{Chord}). Curvature categories are "hard left," "normal left," "flat," "normal right," and "hard right." Deflection categories are "sharp left," "left," "straight," "right," "sharp right." Leg length categories are "short," "medium," "long," "stopped anomaly" (used to indicate double-stamped points), and "jitter anomaly" since radar jitter detection is based on leg length ratios.

2.3.2. Level 2 Processing. Level 2 categorization groups Level 3 nodes according to curvature. Hard left and normal left map to Level 2 category "left." Similarly, hard right and normal right map to Level 2 category "right." Level 3 category flat maps to Level 2 category "straight."

Once processed, any Level 2 node will have between 1 and $n-2$ Level 3 nodes as children. For any Level 2 node, all child nodes must be in the same contiguous sequence. If two Level 3 nodes of the same type are interrupted by a different type, they will be children of different Level 2 nodes.

One may note that mapping Level 3 nodes to Level 2 nodes does not add significant semantic interpretation. The primary purpose of this level of processing is to group like Level 3 nodes into a single Level 2 node. This is necessary because a segment of one kind may include multiple key points which are of the same type.

Level 2 nodes have some geometric properties which are aggregated from the underlying Level 3 geometric information. The most important of these are total point deflection ($\Sigma\delta_{Chord}$) and total leg length. We found that in certain cases partially aliased key points at the boundary between a curved Level 2 segment and a straight Level 2 segment were incorrectly being assigned to the straight segment due to the low value of the transitional curvature. To resolve this issue, a subprocess was added allowing curved Level 2 segments to steal a point from an adjacent straight Level 2 segment. This improved the accuracy of Level 2 curve total point deflection values.

2.3.3. Level 1 Processing. Level 1 categorization groups Level 2 nodes according to multiple patterns in the Level 2 aggregate attributes. Level 1 categories are

1. Cruise: Long, generally straight segments
2. Course Turn: Cruise/Turn/Cruise in which turn $\Sigma\delta_{Chord}$ is less than a U Turn
3. S Curve: A pair of curves in opposite directions with no intermediate straight segment.
4. U Turn: A sequence of straight, turn, straight, in which turn $\Sigma\delta_{Chord}$ is approximately 180°
5. Racetrack: A sequence of turning segments or a single turning segment in which turn $\Sigma\delta_{Chord}$ is greater then 360°
6. Boustrophedon: a sequence containing two or more U Turns in which alternate U turns turn in the opposite direction and straight segments interleave the alternating U Turns
7. No Category: This is the default category and is assigned to Level 1 nodes which do not meet the criteria for other Level 1 categories.

The sequence of Level 1 Categorizations is the primary data product of the CPM. This sequence is viewable in one of the available output types, the parse tree graph. However, the Parse Tree Graph format is not amenable to further analysis. For analysis, the Level 1 Categories are mapped to single characters, which form a string of characters useful for analysis.

2.3.4. Output Files. As described in Section 2, available outputs are

1. a kml file for visualization using Google Earth™. (.kml),
2. a plot of the parse tree graph (.png),
3. a detailed arc data report including derived geometry parameters (.csv),
4. a statistical summary report of Level 1 categorizations (.csv),
5. a Level 1 hash bin report grouping all trajectories by a hash string of the Level 1 categorization. (.csv)

An example of item 1, above, is given in Figure 2.1, which is kml output visualized in Google Earth™. An example of item 2, the parse tree graph, is given in Figure 2.2. No example of Item 3, a detailed arc data report, is provided.

The first three kinds of outputs are time consuming for the analysis process to generate, so these are not generally created for large batches. More commonly, a large run outputs a batch report, data analysis is performed on that report, and a human selects specific trajectories from the batch report of either item 4 or item 5 to be visualized in one or more of the other three reports for further assessment.

Name	Cruise Count	Cruise Percent	Turn Count	Turn Percent
07132202_AAL100	4	81.9%	8	16.1%
07171248_AAL1	3	94.5%	5	3.4%
07230452_AAL10	3	91.5%	5	6.7%
09061717_AAL100	3	74.8%	5	20.5%
08250149_AAH5503	3	67.9%	4	16.3%
07092229_AAL1000	3	66.9%	7	25.2%

Table 2.2
Portion of a Statistical Summary Output table for a test run over a large dataset of trajectories.

Level 1 Category	Single Letter Hash
Cruise	C
Course Turn	T
S Curve	S
U Turn	U
Racetrack	R
Boustrophedon	B
No Category	N

Table 2.3
Hash Mapping from Level 1 Category to a single letter.

The statistical summary report, item 4 above, consists of 15 columns. One column contains a unique name for each trajectory. Subsequent columns are the number of segments of a given category and the percent of total for each of the seven categories. A subset of the summary output is shown in Table 2.2.

2.3.5. The Hash Bin Output Report. The most useful of the available outputs is the Level 1 hash bin report, listed above as item 5. This report groups trajectories together which have the same Level 1 Categorization sequence.

As described in Section 2.3.3, there are six Level 1 categories plus 'No Category.' We map each of these descriptive names for the categories to a single letter as shown in Table 2.3

For example, if a trajectory is found to have Level 1 categorizations of Course Turn, Cruise, and S-Curve, its hash would be TCS. When a batch run generates a hash bin report, it computes the hash for every trajectory, then it bins all like trajectories into groups based on their hash. This results is a report in which every trajectory on a given row has the identical Level 1 Categorizations (including order). Table 3.2 shows a portion of a hash bin report.

3. Results. Software development was carried out in Python primarily on a small number of trajectories. Then batch runs were executed on several large datasets.

The primary question we want to ask is, can we use the Curvature Parsing Method to identify trajectories of interest for further investigation by humans from a large real world dataset? It may be noted that what constitutes a "trajectory of interest" is not being defined in this report since the method allows for so many kinds of analysis to be performed.

3.1. Utilizing the Statistical Summary Output Report. To answer the primary question, we ran several queries on the summary data to see what answers might turn up. Upon flagging trajectories of interest, we plotted the results in kml for human inspection.

One batch run resulted in 2,405 flight trajectories. Only the statistical summary output is available for this run. The first query was to look for trajectories with holding patterns by finding trajectories with a high percentage of Racetrack pattern. After the statistical

Name	Cruise Count	Cruise %	Turn Count	Turn %	No Cat Count	No Cat %	Race-track Count	Race-track %
080118_AAL1	2	68.2%	2	12.9%	0	0.0%	1	18.4%
072818_AAL1	1	34.9%	3	46.1%	2	7.1%	1	13.2%
082214_AAH1	1	77.4%	1	7.5%	0	0.0%	1	12.2%
072517_AAL1	3	42.2%	3	41.9%	1	2.9%	1	9.9%

TABLE 3.1
Portion of Summary Output Table sorted by Racetrack Percent.

summary output cvs was generated, we sorted the values so that Racetrack appeared at the top of the list.

After reviewing these four flights visually using kml output, we found that one of them, 082214_AAH1, does not contain any racetrack segments. Investigation as to why it incorrectly reports to have 12% racetrack is left as future work.

The other three appear all to be the same flight from Charlotte to JFK, but on different days. In the dataset under study, these three had noteworthy racetrack segments as indicated by the statistical summary output, but the holds do not occur near the destination as is the usual case. Rather they were all near the midpoint of the trajectory over southeastern Virginia. A detail of one of these flights, 072818_AAH1, is shown in Figure 3.1.

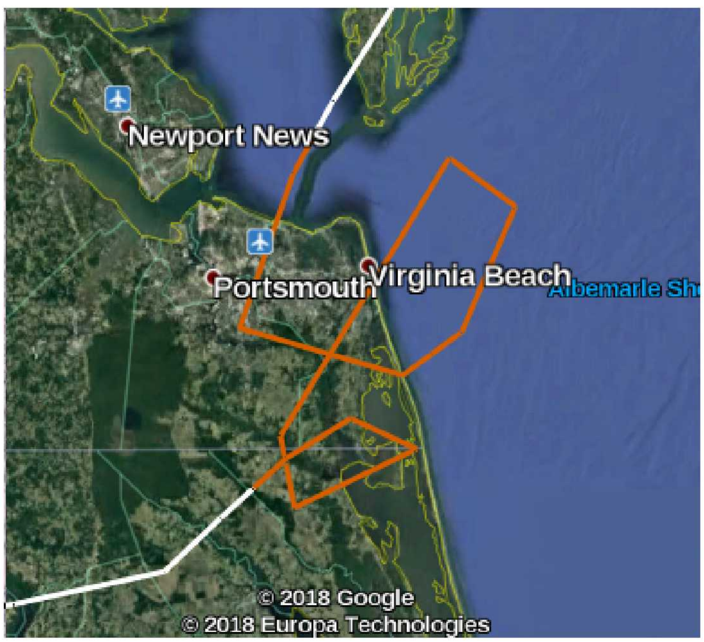

FIG. 3.1. *Holding pattern for a flight from Charlotte to JFK. The uninterrupted turning in the same direction is categorized as a Racetrack segment. The trajectory was identified from the statistical summary report sorted by Racetrack Percentage.*

3.2. Utilizing the Level 1 Hash Bin Output Report. We carried out a second batch run on a different dataset. In this run we generated a Level 1 Hash Bin report. A total of 21,985 trajectories were processed in 339 seconds resulting in 886 distinct hash combinations. When the report is sorted in descending order of trajectory count per hash, the partial result is shown in Table 3.2. The most common Level 1 categorization hash string is 'C' (Cruise only), with 3,982 trajectories in this category. Not shown in Table 3.2, there are 518 singleton hash strings, which are hash strings with only a single trajectory.

Hash String	Trajectory Count	Trajectory IDs
C	3,982	08302100_2XSQM, etc
TC	2,205	09222020_A459, etc
CT	2,114	09041341_9ZHRA, etc
T	1,814	09180011_A5A775, etc
TCT	1,639	09012217_A111971, etc
CTC	1,249	08211613_A1L1958, etc
CTCT	865	08270021_A1, etc
TCTC	660	08030648_AAH11, etc
TCU	462	07141358_AAL100, etc
TCTCT	407	09191922_0UXUB, etc

TABLE 3.2
Portion of a Hash Bin Output Report sorted by Trajectory Count.

No further data assessment was carried out on the Level 1 Hash Report (L1HR) results. However, one may note that it is in this Hash Report that a type of trajectory similarity emerges. Unlike the statistical summary report, the L1HR preserves the sequence of categories, so the sequence constitutes part of the trajectory similarity. The L1HR aggregates and reduces the trajectory data to a form which is better suited for some established data mining approaches.

4. Discussion. The question posed in the Results section has been answered in the affirmative. We are indeed able to use the Curvature Parsing Method to identify trajectories of interest for further investigation. The research into this approach is a partial success. We can get Level 1 Categorizations and perform analysis on these results which help us understand a large dataset, find similarities among temporally and spatially distant trajectories, or quickly find trajectories with specific characteristics.

We claim that this success is only partial because of certain shortcomings. First, we find a number of trajectories which are being classified incorrectly. There may be other bugs in the software which may similarly be considered implementation shortcomings and would be resolved with further work. Another kind of implementation shortcoming is that course correction turns in the middle of long cruises are not being identified as turns.

There are also shortcomings related to the process which must be refined or mitigated. Specifically we refer here to concepts related to curves being missed or incorrectly characterized due to the inherent limitations of sample rate.

Given these observations, we believe a potential for the Curvature Parsing Method to be of value is present. As developed thus far, the Curvature Parsing Method shows promise, but it should be developed more. The Level 1 semantic segments are reasonable and enable an analyst to find alignments with certain features of interest from a large dataset.

4.0.1. Future Research. One indication of the success of the project is that there are multiple directions that future research could take. These directions may be grouped into three themes: Geometry, Categorization, and Analysis.

1. Geometry
   (a) Continue investigation of how to get the most correct arc values for under sampled curves.
   (b) Determine how to find the exact point of curvature (End Tangent/Begin Curve) given edge aliasing tends to obscure these points.
   (c) For racetrack patterns, find the center point which the craft is orbiting. This information could be used to identify Surveillance flights.
   (d) Determine what the best way is to identify radar jitter and the stopped anomaly, and consider improved approaches to ignore or categorize them.

   (e) Add altitude and speed analyses to gain greater understanding of the trajectory.
2. Categorization
   (a) Investigate whether there should be an additional, higher level of categorization. This may be the case given that Boustrophedon depends on a certain sequence of two other Level 1 categories, cruise and u-turn.
   (b) See what other Level 1 categorizations should be added. It would be advantageous to be able to distinguish "loops" from course turns by finding turns between 181° and 360° (similar to a loop ramp at an interchange). It would also be useful to distinguish between "figure eight" and other patterns.
3. Analysis
   (a) Identify useful types of analytic questions which may be asked of the data products.
   (b) What are common patterns in different situations. For example, how often do S-curves appear within the final three category segments, thus indicating a runway alignment maneuver.
   (c) How can machine learning be applied to these processes either to detect previously undetected patterns, or to improve the selections of threshold values at all categorization levels.

5. Conclusion. The Curvature Parsing Method is an innovative approach to identifying and analyzing meaningful subsegments of loosely constrained trajectories. Although it needs more development, we believe that the prospective benefits of this method warrant the additional work.

REFERENCES

[1] S. C. Ahearn and S. Dodge, *Recursive multi-frequency segmentation of movement trajectories (remus*, Methods in Ecology and Evolution, 9 (2018), pp. 1075–1087.

[2] D. J. Findley, J. E. Hummer, W. Rasdorf, and B. T. Laton, *Collecting horizontal curve data: Mobile asset vehicles and other techniques*, Journal of Infrastructure Systems, 19 (2013), pp. 74–84.

[3] T. Hickerson, *Route location and design*, McGraw-Hill, 1967.

[4] W. Luo and L. Li, *Automatic geometry measurement for curved ramps using inertial measurement unit and 3d lidar system*, Automation in Construction, 94 (2018), pp. 214 – 232.

[5] B. D. Newton, M. D. Rintoul, C. G. Valicka, and A. T. Wilson, *Center for computing research summer research proceedings 2015.*, SUMMER PROCEEDINGS 2015, The Center for Computing Research at Sandia National Laboratories, (2015), pp. 202–211.

[6] W. Rasdorf, D. Findley, C. Zegeer, C. Sundstrom, and J. Hummer, *Evaluation of gis applications for horizontal curve data collection*, Journal Of Computing In Civil Engineering, 26 (2012), pp. 191–203.

[7] M. D. Rintoul and A. T. Wilson, *Trajectory analysis via a geometric feature space approach*, Statistical Analysis and Data Mining: The ASA Data Science Journal, 8 (2015), pp. 287–301.

[8] M. D. Rintoul, A. T. Wilson, C. G. Valicka, W. P. Kegelmeyer, T. M. Shead, B. D. Newton, and K. R. Czuchlewski, *Panther. trajectory analysis*, (2015).

[9] P. T. Schrum Jr., *Get the kinks out determining roadway radii from polylines*, NC GIS Conference. Time Index 14:30 – 19:19.

[10] C. E. Shannon, *Communication in the presence of noise*, Proceedings of the IEEE, 86 (1998), pp. 447–457.

[11] *Aircraft Situation Display To Industry: Functional Description and Interface Control Document*, tech. rep., Volpe Center, 2000.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.