SAND2014-1360C

Trajectory Analysis via a Geometric Feature Space Approach

Mark D. Rintoul
Sandia National Laboratories
Albuquerque, New Mexico
mdrinto@sandia.gov

Andrew T. Wilson
Sandia National Laboratories
Albuquerque, New Mexico
atwilso@sandia.gov

Randy C. Brost
Sandia National Laboratories
Albuquerque, New Mexico
rcbrost@sandia.gov

ABSTRACT

We want to organize a body of trajectories in order to identify, compare and classify both common and uncommon behavior among objects such as aircraft and ships. Existing comparison functions such as the Frechet distance are computationally expensive and yield counterintuitive results in some cases. We propose an approach using feature vectors whose components represent succinctly the salient information in trajectories. These features incorporate basic information such as total distance traveled and distance between start/stop points as well as geometric features related to the properties of the convex hull, trajectory curvature and general distance geometry. Most of these geometric features are invariant under rigid transformation. We demonstrate the use of different subsets of these features to identify trajectories similar to an exemplar, cluster a database of several hundred thousand trajectories, and identify outliers.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*; I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement—*Feature representation*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Tracking*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*

General Terms

Algorithms, Measurement, Verification

Keywords

Trajectory, Flight

1. INTRODUCTION

The growth of remote sensing capabilities has resulted in a well-documented explosion of image data[4]. However, interpretation of that data mostly remains a human activity. One of the important changes in this revolution of sensing data is not only increasingly high-resolution and wide-field imagery, but also rapid time sampling. This data presents an interesting computational analysis problem that is inherently different than that associated with comparing images for large, durable feature changes. With multiple data points, one can track particular objects, and build a series of time stamped location points that make up a trajectory. Of course, the problem of comparing trajectories is not limited to traditional overhead imagery analysis. Animal tracking is of great interest to the biology community to understand behavior[9]. In general, any multidimensional dataset that has time stamped points can be considered a trajectory through phase space.

One example of a difficult but important example of intent that we have chosen to study is classifying aircraft behavior based on their trajectories. It is important for a number of reasons. First, there are a number of obvious security reasons. It is useful to comb data to search for criminal or terrorist activity. Understanding patterns of both normal and anomalous behavior is critical to optimizing public air traffic resources. Obtaining details of airline performance that are not usually called out is also a potential application.

The aircraft intent problem also has the quality of having a more complicate space of input and output. Generally the input consists of time-stamped location and altitude data from which other derived quantities such as speed and heading can be calculated to a certain accuracy. In many cases this input is derived from multiple data sources and has many errors and omissions. The outputs are dependent on the problem of interest. This could include looking for regular patterns, anomalous patterns, patterns that correspond to a specific behavior, clustering into groups or finding a flight similar to an input trajectory. The outputs described above are not necessarily well-defined and in some cases have a human-defined component to them. The net result of these complexities is a potentially rich set of ways to go about building the model that connects the inputs and outputs.

There have been a number of approaches to the trajectory problem that include Fourier descriptors[1], earth mover distance[3], hidden Markov Models[2], Hausdorff-like distances[7], Bayesian models[8] and other approches. We propose an alternative approach that is primarily focused around geometric quantities related to the track as a whole. These quantities have many desirable properties. First, most describe the trajectories as a whole, and this appears to correspond better to how a person views the trajectories. Second, most of these descriptors correspond to values that can be pre-calculated and compared quickly with other trajectories, as opposed to comparison measures that require specific comparisons between every track with every other track. This would allow rapid lookup in a database. Finally, for many practical questions of interest that separate flight behaviors, these geometric descriptors correspond fairly closely to one or more quantities that describe the behavior of interest.

In this paper, we begin by describing some of the related work that has been done in the area of comparing trajectories, in terms of both aircraft and more general work. In Section 3 we move on to describing more carefully the specific problems we are trying to solve by designing geometric measures for aircraft trajectories. Section 4 gives a summary of the quantities that were evaluated as geometric measures. The techniques used to find trajectories in relevant feature spaces, and the relationship to clustering is described in Section 5. The evaluation of the quality of the different geometric measure is given in Section 6, and then we summarize our work and offer suggestions for future work in Section 7.

1.1 Notation

We will use the following notation when describing trajectories and their features.

- A trajectory $\mathbf{T}$ comprises $n + 1$ timestamped points $(x_0, t_0), (x_1, t_1), \dots, (x_n, t_n)$.
- Given $\mathbf{T}$, angle θ_i is the turning angle from vector $(x_i - x_{i-1})$ to $(x_{i+1} - x_i)$. Informally, θ_i is the turn between segments i and $i + 1$ in the trajectory. Positive angles indicate counterclockwise turns.
- $|\mathbf{T}|$ is the total length of all the segments of $\mathbf{T}$.
- $||x_n - x_1||$ is the *end-to-end distance* of $\mathbf{T}$.
- $\mathcal{C}(\mathbf{T})$ is the convex hull of the points in $\mathbf{T}$. Points $c_1, c_2, \dots, c_m \subset x_1, \dots, x_n$ form the vertices of $\mathcal{C}(\mathbf{T})$.
- $\overline{\mathcal{C}(\mathbf{T})}$ is the centroid of $\mathcal{C}(\mathbf{T})$.

2. BACKGROUND

2.1 Previous Approaches

The fundamental computer science issues related to comparing two trajectories have been studied for many decades in their most general form. If one considers a trajectory, $\mathbf{T} = \{(\mathbf{x_1}, t_1), \dots, (\mathbf{x_n}, t_n)\}$ to simply be a set of points in an $D + 1$-dimensional space, there are a significant number of application drivers outside of aircraft trajectory comparison. These include object recognition, handwriting analysis, and many different forms of time-series analysis.

There have been many different distances defined to measure how far apart two trajectories are. Perhaps the most straightforward measure of distance between two curves is the Hausdorff metric. For two trajectories, $\mathbf{A}$ and $\mathbf{B}$, it is defined as greatest distance out of all of the distances between from a point on $\mathbf{A}$ to the nearest point on $\mathbf{B}$. This gives a rough sense of the distance between two curves, but doesn't take into account the direction traveled on the trajectories.

One of the most well-known metrics associated with curve similarity that does take the direction into account is the Frechet distance. The Frechet distance $F(\mathbf{A}, \mathbf{B})$ is formally defined as

$$F(\mathbf{A}, \mathbf{B}) = \inf_{\alpha,\beta} \max_{t \in [0,1]} \{d(\mathbf{A}(\alpha(t)), \mathbf{B}(\beta(t)))\} \qquad (1)$$

where $\alpha(t)$ and $\beta(t)$ are continuous, non-decreasing reparameterizations of $\mathbf{A}$ and $\mathbf{B}$, respectively, onto the interval $[0, 1]$. Eiter and Mannila[6] have extended this definition in a straightforward manner to the case where $\mathbf{A}$ and $\mathbf{B}$ are described by discrete points as polygonal curves. Both variations of the Frechet distance represent the minimum length of a leash required for a man, following one curve, to walk a dog that is following the other curve.

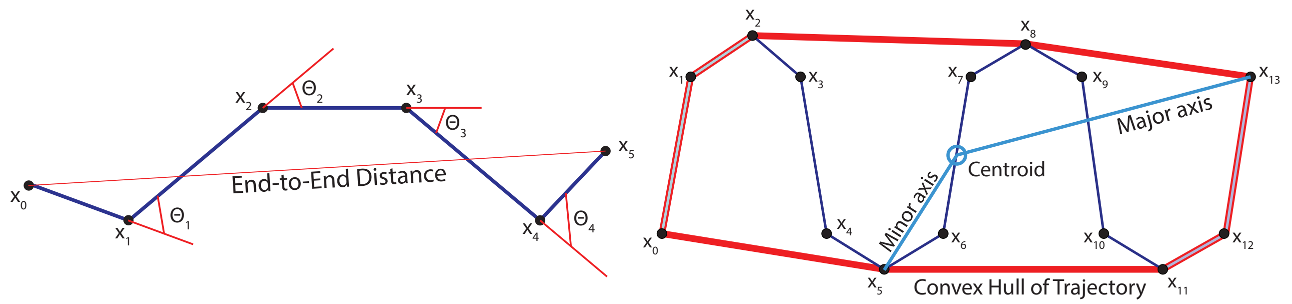
One problem that both the Hausdorff distance and Frechet distance have is that they do not allow for translational, rotational or reflectional invariance. If they do not naturally represent curves that would naturally be positioned perfectly, they must be *aligned* for the those two distances to compare the shapes. Proper alignment is a difficult problem, and typically one would have to do a Procrustes type of analysis to align them[5], or alternative methods based on dynamic time warping or sophisticated edit distance approaches that try to match geometric distance and curvature between points. Additionally, hidden Markov models have also been used to try to compare and classify trajectories.

2.2 Why Something Different?

The metrics described above were primarily designed to do one-on-one comparisons between two trajectories, but for very large-scale work in identifying behavior in trajectories ($> O(10^6)$ trajectories), they become difficult to work with. Many of these distance metrics have a behavior that is $O(ab)$, where a and b are the number of discrete points in the trajectory. Furthermore, there is little that can be pre-computed for each individual trajectory, so every comparison calculation must be done completely for each comparison of interest. What would be ideal is a way to measure similarity that:

- Can be calculated once for each trajectory.
- Can be calculated for each trajectory in a time that is linear in the number of trajectory points.
- Can used to calculate similarity between two trajectories in constant time.
- Can be used to cluster trajectories.
- Has translational, rotational, and potentially scaling and reflection invariance properties.
- Is based on characteristics of the trajectories that can effectively categorize behavior.

The approach here is to use simple scalar measures associated with each trajectory (such as time, total distance, etc.), and combine those values with *geometric* scalar quantities that describe the relevant geometric characteristics of the trajectory. This gives us a feature vector associated with each trajectory that can be used to store information about, and do comparisons between different trajectories. These comparisons between feature vectors can be done through a specifically defined vector product that can be done in a



(a) (b)

Figure 1: Illustration of the parts and properties of a trajectory that we use to compute features. A trajectory T comprises $n+1$ points $x_0, x_1, \ldots, x_n$. In (a) we see a trajectory T labeled with its vertices x_i, turning angles $\theta_1 \ldots \theta_{n-1}$ and end-to-end distance $||x_5 - x_1||$. In (b) we see another trajectory U with vertices $x_0 \ldots x_{13}$ and convex hull $\mathcal{C}(\mathbf{U})$. We approximate the aspect ratio of $\mathcal{C}(\mathbf{U})$ as the ratio of the lengths of its major and minor axes where the major axis connects the centroid of $\mathcal{C}(\mathbf{U})$ with the most distant vertex x_i and the minor axis connects the centroid with the nearest vertex x_j.

time that is constant with respect to the length of the trajectories themselves. These features can also be used in traditional databases or specially-designed database machines to do lookups very quickly on very large databases.

3. PROBLEM DEFINITION

We define here more precisely what is meant by trajectory comparison. There are a few different types of problems that involve trajectory comparison. Some of the more important ones that we will cover are

- Can you find the trajectories in a database that are most similar to a given trajectory?
- Can you find trajectories that exhibit a behavior of interest that can be translationally, rotationally or scale invariant?
- Can you divide trajectories into specific clusters?
- Can you find trajectories that are outliers with repsect to a given set of trajectories?

In order to solve these problems using the geometric feature vector approach, we have to define the quantities that will be useful to construct the feature vector. These fell into a few different categories that are described below.

3.1 Distance Measures

These measures include many straightforward measure associated with the flight and include

- End-to-end distance of the flight:

$$d_e(\mathbf{T}) = ||x_{n+1} - x_1||$$

- Total distance traveled (length of trajectory):

$$d_t(\mathbf{T}) = \sum_{i=0}^{n} ||x_{i+1} - x_i||$$

- Distance from a given fixed point or set of points
- Centroid of points:

$$\overline{\mathbf{T}} = \frac{1}{n} \sum_{i=0}^{n+1} x_i$$

The first two of these measures are simple but important ones for characterizing flights, while the third can be calculated for more specific concerns related to relevant fixed points on the ground. Note that the fourth, along with similar measures defined later, consist of two values defined on the surface of a sphere (usually by longitude and latitude) and not just a single value.

3.2 Heading Measures

One can also define measures associated with how straight a flight is, such as

- Total curvature:

$$c_{total}(\mathbf{T}) = \sum_i \theta_i$$

- Total turning:

$$c_{abs}(\mathbf{T}) = \sum_i |\theta_i|$$

- Average curvature/turning:

$$\frac{1}{n} c_{total}(\mathbf{T}), \frac{1}{n} c_{abs}(\mathbf{T})$$

These measures turn out to be very useful either by themselves or in conjunction with other measures, to separate out different types of flights.

3.3 Geometrical Measures

These more sophisticated measures often say more about the shape of the flight than the more basic measures listed above, and are key to some of the results later in the paper. These measures include

- Area covered by flight, defined here as the area of the convex hull of the flight points

- Aspect ratio of the convex hull of the flight. This is defined as the ratio of the shortest to the longest axis of the polygonal convex hull of the points. (FOOTNOTE: We approximate the length of the shortest axis as

$$2 \min_i ||\overline{\mathcal{C}(\mathbf{T})} - x_i||$$

where x_i is on the convex hull of $\mathbf{T}$. The length of the longest axis is similarly

$$2 \max_i ||\overline{\mathcal{C}(\mathbf{T})} - x_i||$$

for x_i on $\mathcal{C}(\mathbf{T})$.

- Length of the perimeter of the convex hull
- Centroid of convex hull $\overline{\mathcal{C}(\mathbf{T})}$
- Ratio of end-to-end distance traveled to total distance traveled:

$$\frac{d_e(\mathbf{T})}{d_t(\mathbf{T})}$$

. This will never be greater than 1.

- Radius of gyration of the points (FIXME: need math here)

We also believe that the geometric measures described above seem to capture more holistic views of the trajectories and correspond closely to how humans view the trajectories. However, this work will not examine the question in details, and detailed comparisons to human studies will be left to a future work.

There is a final geometric measure based on the concept of *distance geometry* that we will use that describes complex shapes in more detail. First, parameterize a trajectory uniformly over the interval $0 \leq t \leq 1$. Then choose a set of m intervals (t_{m1}, t_{m2}) and measure those distances. This set of m values then can be used as geometric measures to describe the shape of the trajectory. These m values represent a geometric measure that is invariant to translation, rotation and reflection. However, if you normalize these m values by the largest value, such that all of the values are between 0 and 1, one obtains a measure that is also *scale invariant.*

3.4 Use of Feature Vectors

The feature vector approach allows two different approaches to solve the problems listed above. The first one is the most straightforward. One can calculate the feature vectors and then use traditional searching or clustering approches algorithms using a distance metric defined by the feature vectors.

However, there is another approach that turns out to be faster and more general for some applications. If one chooses the feature vector carefully and builds a distance metric on those vectors that is expressible as an L^p norm, then one can use a spatial indexing scheme such as an *r-tree* to store feature vector values, search for nearest neighbors, and even do clustering.

4. RESULTS

4.1 Data

The data used to test our algorithms and generate the results is the ASDI (Aircraft Situation Display to Industry) data set that includes most US civilian air traffic that have flight plans on file. This is the same data set that is used by many flight status web sites. It originates from the FAA (Federal Aviation Administration) and comes from multiple original sources. We get the data from AirNav, Inc., which does some structuring of the data and puts it in an XML format.

The data consists of approximately 50,000 flights per day, with approximately 6 months of flights represented. Each flight consists of a flight ID, position (latitude and longitude) data and a time stamp on each data point along with a large amount of supporting metadata. The number of data points represented by each flight ranges from less than 10 to many hundreds. The data points for each flight generally were spaced approximately 60 seconds apart, but could spaced more closely (or even have the same time stamp) if the data was obtained from multiple sources. The metadata associated with each point could include, but doesn't always, altitude, speed, heading, departure/arrival airport, etc. If one is working on an application where there is significant metadata available, it certainly makes sense to include it to help classify flights. However, the focus of theis work is study how geometric classifiers can be used to compare flights and so the associated metadata for each point was not used.

The data points did not arrive sorted by flight, but were instead sorted by time. Separation into individual flights was done via sorting by flight ID, and then looking for potentially large time breaks between points that would indicate a landing and take-off from a flight that included multiple stops under a common flight ID. In general, a gap of 30 minutes between points was indicitive of a landing/take-off, although values between 20 and 60 minutes didn't change the flight separation process significantly.

Each flight was checked for data irregularities before it was finally accepted as a proper flight. Mostly, this consisted of checking to see if a point was an unreasonable distance away from its neighboring points given the time separation between them, and removing them. In general, we required this effective speed to be approximately 3-10 times faster than a typical airplane in order to remove only the especially bad points. The reason for this is that we were interested in testing our measures against data that potentially had significant uncertainty in the position in order to see how robust they were in the presence of noise.

4.2 Simple Geometrical Filtering

The first examples we show here are just primarily intended to test some of the more straightforward aspects of geometric search and are based around single passes through data sets looking for specific values of parameters that represent a given type of behavior.

4.2.1 Avoiding Airspace

One possible question that could be asked of a collections of flights is, "Is there a section of airspace that flights seem to be avoiding?" A geometric signature corresponding to such a question could be described in a number of ways. A simple way would be to look at flights that traveled a significant distance (in order to exclude flights that are simply

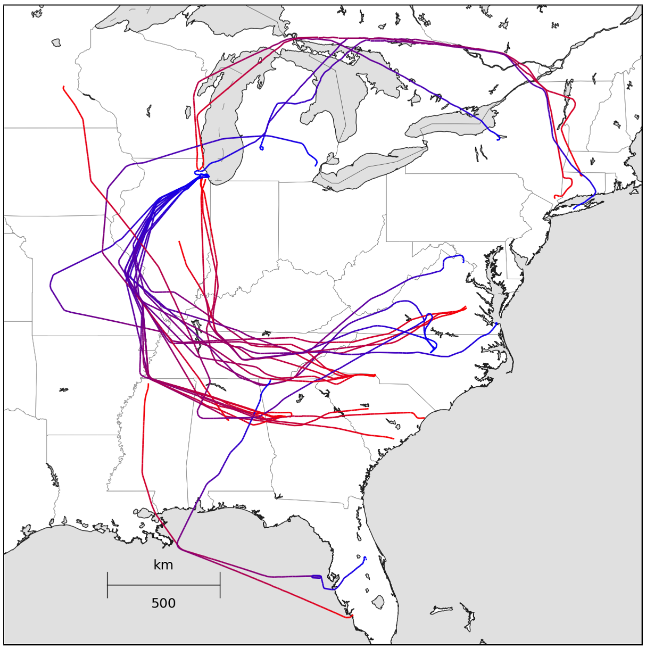


Figure 2: Examples of flights found for the "avoiding" specification. In this case, we required the end points of the flight to be at least 1000 kilometers apart, the ratio of the end-to-end distance of the flight to the total flight distance to be < 0.7, and the aspect ratio of the convex hull to be $> \frac{1}{3}$.

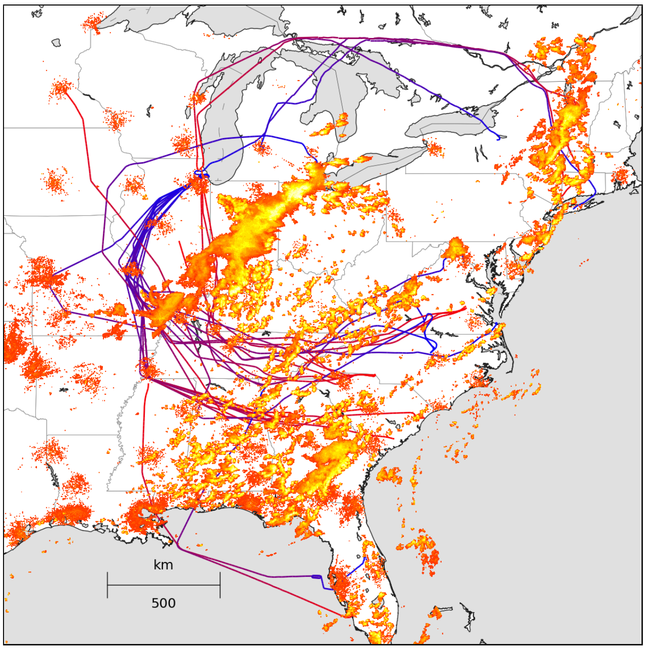


Figure 3: Most of the trajectories identified with the "avoiding" specification were responding to this event: a severe weather system crossing Illinois and Indiana. This weather map was captured at 2:30PM Eastern Daylight Time (UTC-5) on July 10, 2014.

flying circles as part of training), but traveled a distance that was significantly larger, but not too much larger, than the distance between their take-off and landing points. Furthermore, to exclude flights that simply meander, one could put a constraint on the size of the convex hull of the flight, relative to the length. Flights from July 10, 2013 satisfying those criteria area show in Figure 2. Upon doing a little digging, one finds that there was a large cell of thunderstorms in the central Indiana area that evening that the flights were all avoiding. Figure 3 shows a weather map from midday overlaid on those same trajectories.

4.2.2 Holding Pattern

Another somewhat distinctive pattern that one might be interested in is that of a holding pattern, indicitive of a plane that has flown for a distance, but is now flying in circles due to some sort of landing delay. The geometric constraints for this search are having a flight of some moderate length, but a signficant total curvature that would be unusual for a point-to-point flight. This was successful in the sense that it returned many flights that had clearly been instructed to circle while waiting permission to land. We decided to try to make this more difficult and find flights that appeared to be in a holding pattern and then diverted to a different airport. This was done by requiring a flight to have significant turning, but also cover lots of distance and have a significant aspect ratio to its convex hull. The flight that was found is shown in Figure 4. Examining the metadata shows that it was indeed inbound to Atlanta and was diverted to Chattanooga in early June of 2013.

4.2.3 Mapping Flights

Given the advances in imaging technology and the burgeoning business in on-line map services, there are a significant number of planes flying in a back-and-forth, or boustrophedon, pattern. This type of flight will have a significant length, but the actual flight will be enclosed by a fairly compact shape. For this search, we require a reasonably long total distance, but a small radius of gyration. This gives a number of false positives, but it also finds the flights shown in Figure 5. That was just a sample of the flights found. There were mapping flights all over the country, and the search found approximately 10% "false positives" that did not seem to correspond to mapping flights. For the sake of testing, we also implemented a "feature" that did a simple search for straight segments separated by 180 degree turns that also did well, but was brittle with respect to minor variations in the mapping process.

5. DISTANCE GEOMETRY EXAMPLES

As an example for the use of distance geometry technique that was described in section 3, we will use one of the flights that was found above in the example on finding flights that appear to be avoiding a specific area of airspace. While the goal in that example was achieved by attempting to describe the features of interest that would find the flights, using distance geometry, we can do something even simpler. By starting with flight shown in Fig 6a we can measure distances at various points along the flight, and build a feaure vector with the distances normalized by the largest distance. We can then compare the distance between that feature vector and the feature vectors from the other flights in the database using an L^2 distance to find flights that have a similar shape.

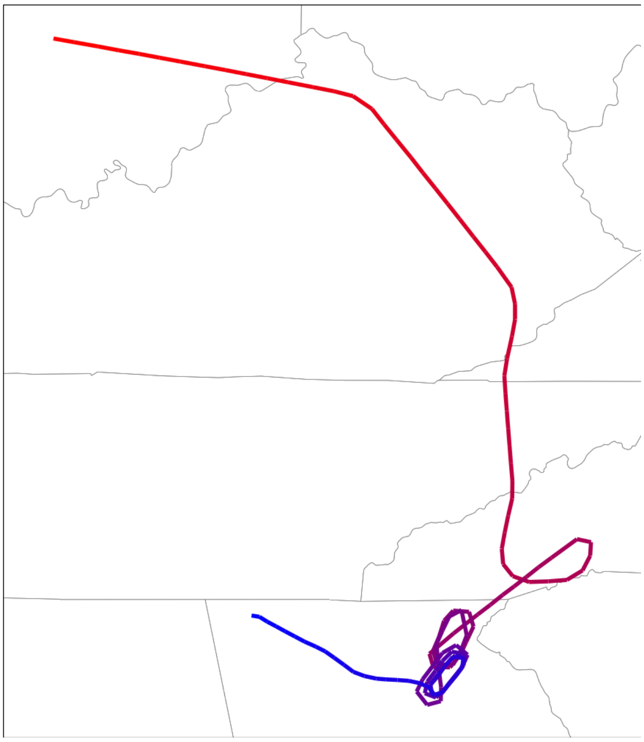


Figure 4: Examples of flights found for the "holding and diverted" specification. In this case, we required the end points of the flight to be at least 200 kilometers apart, the total amount of turning to be $> 20\pi$ radians, and the aspect ratio of the convex hull to be $> \frac{1}{10}$.

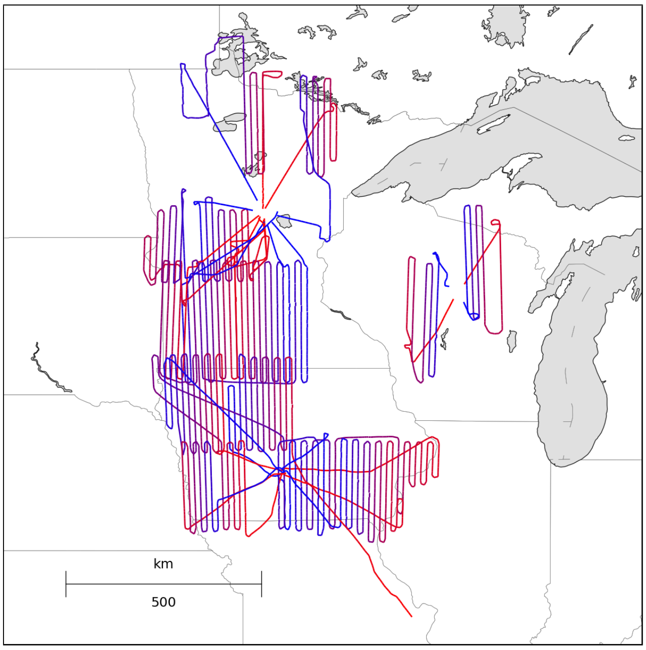


Figure 5: Examples of flights found for the mapping criteria. We require the flights to be longer than 800 miles, and have a convex hull aspect ratio greater than 1/8. We then took the top 50 flights in terms of having a small ration of radius of gyration to total distance flown. A small sample of the results are shown below. There were approximately 10% false positives for these criteria.

In our example, we chose 10 different distances to use as the intratrajectory distances. If the curve was parameterized as a function of time over $[0, 1]$, we approximately chose the full distance ($d(0, 1)$), the two half distances ($d(0, \frac{1}{2})$ and $d(\frac{1}{2}, 0)$), and so on through the 4 quarter distances. While we could have estimated the distances at the precise time points through interpolation between the discrete points, we simply chose the points that was closest to the interval boundary under the assumption that the points were roughly equally spaced. This made the lookup very fast and did not significantly change the outcome from where we wrote a routine to do a careful parameterization of the time and found precise distances through interpolation between points.

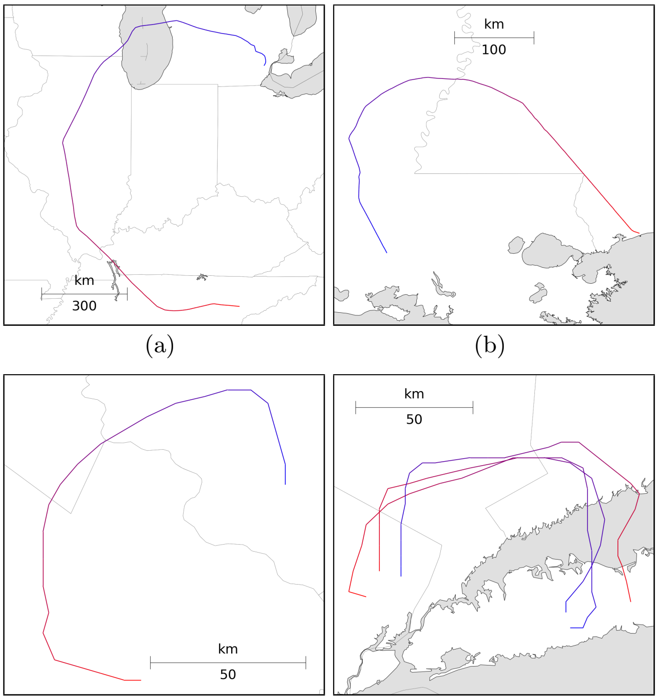
The results for that comparison are also shown in Figure 6. There were a wide variety of results, with quite a few different sizes and orientations, but all fundamentally shaped the same. We had also originally tried to do these comparisons with curve alignment algorithms that were based on dynamic programming techniques, but they took much longer and didn't match the global nature of the curves due to their focus on matching local structures.

6. INDEXING WITHIN FEATURE SPACE

The previous example that used distance geometry found trajectories that were similar to a given trajectory by calculating a feature vector of intertrajectory distances for the initial trajecory and doing a serial comparison to all of the flights in the database. However, for a large database, multiple searches would each individually take a time proportional to the number of flights in the database due to the need to compare each feature vector against the flight of interest. An alternative to this is to create a data structure that will create a spatial index within the multidimensional feature space that will allow searching only flights that are nearby for similarity.

One of the most popular data structures for this type of spatial indexing is a *R-tree*. The R-tree is a multidimensional data structure that represents objects by their minimum bounding n-dimensional rectangle in the next highest level of a tree. This hierarchical structure allows for logarithmic time search and insertion. If the specific characteristics required for comparison are known a priori, a multidimensional space of those geometric features can be populated with the database of flights, and finding "similar flights" becomes a neighbor search that is simple to do on the R-tree.

As an example of this, we demonstrate a somewhat more sophisticated search. We start with the flight shown in Figure 7, a roughly figure-eight shape which is somewhat unusual among the flights in our database. This flight is somewhat harder to write the descriptor for. Instead of writing the descriptor, we'll just define the different dimensions of the feature space to be features that we guess will be relevant. For this test, we simply chose 3 features: the total distance, the ratio of the end-to-end distance to the total distance, and the aspect ratio of the convex hull. We did



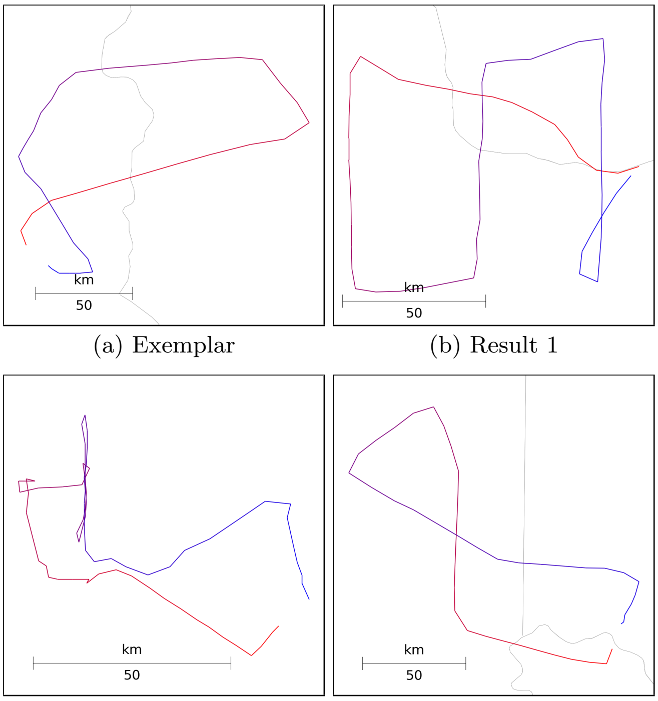
(a) (b) (c) (d)

Figure 6: Examples of curve matching using the distance geometry algorithm. The curve to be matched is shown in (a). Two examples of matched curves are shown in (b) and (c), though at very different scales as (a). The curve in (b) flies around the southern Louisiana area, while the curve in (c) flies around Washington, DC area. Finally, in (d), we see some examples of a pattern of flying around NYC, in both directions.

a search over approximately 50,000 flights (about 1 day's worth), and asked for the 10 closest points in feature space. Three of the closest points are shown for comparison. Given the small dimension of the feature space, some of the other neighbors did not resemble the figure-eight shape as well. On an interesting note, you can also search for the flights that are "furthest" away from the test flight above. In this case, the 10 flights furthest away were all long, straight trans-Atlantic flights.

Additionally, embedding the data in the feature space allows clustering to be done in a number of different ways. There are a number of traditional dimensionality reduction techniques that project data down from a high dimensional space to a two-dimensional space so that clusters can be found through visual inspection or by existing algorithms.

Finally, with the feature space embedding, there is a somewhat elegant solution to a difficult problem: finding trajectories that are outliers with respect to a set of other trajectories. Through the feature space embedding method, one can search for individual trajectories, or small clusters of trajectories, that do not have many nearby neighbors. This gives a quantitative definition of the notion of an outlier or outliers with respect to a set of trajectories and their respective features.



(a) Exemplar (b) Result 1 (c) Result 2 (d) Result 3

Figure 7: Examples of curve matching using feature space search. The curve to be matched is shown in a). The dimensions in the feature space here represent total distance, the ratio of total distance to end-to-end distance, and the aspect ratio of the convex hull. The 10 nearest-neighbor points in the feature space were searched for, and 3 of the results are shown.

7. CONCLUSIONS

The key conclusion that we have drawn from the work that we have done is that for many cases, working in feature space, rather than the physical space of the trajectories, is a much more effective way of finding trajectories that match a given set of criteria than using dynamic programming approaches that do more local comparisons. This is partially due to computational issues, but very prelimary discussion have also indicated that these more global geometric features also generally correspond better to how people see trajectories. This is also more aligned with our overall goal of building a tools for analysts to use to find trajectories that correspond to specific *behaviors* and not necessarily to well-defined numerical qualities.

We anticipate that follow-on work will focus on two general areas. The first one will be centered on computational improvements that include implementation on a database machine, a more thorough analysis of the information content in the different features, and examination of more efficient ways to break up the trajectories into segments to find smaller features. We also would like to do experiments with analysts to understand better how people currently compare trajectories using only their experience as a tool.

8. ACKNOWLEDGMENTS

The authors would like to acknowledge the Sandia LDRD program for their support of this work. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

9. REFERENCES

[1] R. Annoni and C. Forster. Analysis of aircraft trajectories using fourier descriptors and kernel density estimation. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1441–1446, Sept 2012.

[2] F. I. Bashir, A. A. Khokhar, D. Schonfeld, S. Member, and S. Member. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Trans. Image Process*, 2005, 2007.

[3] F. Boem, F. A. Pellegrino, G. Fenu, and T. Parisini. Multi-feature trajectory clustering using earth mover's distance. In *CASE*, pages 310–315. IEEE, 2011.

[4] D. I. Daily. Too much information: Taming the uav data explosion. *Defense Industry Daily*, May 16 2010.

[5] I. L. Dryden and K. Mardia. *Statistical shape analysis.* Wiley series in probability and statistics: Probability and statistics. J. Wiley, 1998.

[6] T. Eiter and H. Mannila. Computing discrete frechet distance. Technical Report CD-TR 94/64, Technische Universitat Wien, April 1994.

[7] B. Guan, L. Liu, and J. Chen. Using relative distance and hausdorff distance to mine trajectory clusters. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(1):115–122, 2013.

[8] J. F. P. Kooij, G. Englebienne, and D. M. Gavrila. A non-parametric hierarchical model to discover behavior dynamics from tracks. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 270–283, Berlin, Heidelberg, 2012. Springer-Verlag.

[9] T. A. Patterson, L. Thomas, C. Wilcox, O. Ovaskainen, and J. Matthiopoulos. State-space models of individual animal movement. *Trends in Ecology & Evolution*, 23(2):87 – 94, 2008.