SAND2014-15008C

Trajectory Analysis via a Geometric Feature Space Approach

Mark D. Rintoul*
Sandia National Laboratories, Albuquerque, NM
and
Andrew T. Wilson†
Sandia National Laboratories, Albuquerque, NM

June 9, 2014

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 Fréchet 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.

Keywords: Trajectory, Flight, Feature Vectors, Clustering

CoDA 2014 *Statistical Analysis and Data Mining* tex template (do not remove)

*mdrinto@sandia.gov
†atwilso@sandia.gov

1 Introduction

The growth of remote sensing capabilities has resulted in a well-documented explosion of image data[1]. However, interpretation of that data mostly remains a human activity. In recent years we have seen rapid growth not only in image resolution and field of view but also in sampling frequency. This enables an interesting computational analysis problem – trajectory analysis – that is inherently different than the search for large, durable feature changes. Given multiple data captures we can track particular objects, extract their locations, and build up a series of time-stamped positions that compose a trajectory[2].

Of course, the problem of trajectory analysis is not only of interest in the setting of overhead image analysis. The biology community uses it to examine animal behavior[3]. Molecular dynamics researchers use the trajectories of atoms and molecules to study the behavior and conformations of proteins and polymers[4]. In general, any multidimensional data set that has time-stamped points can be considered a trajectory through phase space.

One example of a difficult but important example that we have chosen to study is the classification of aircraft behavior based on flight trajectories. This problem 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 is also a potential application.

The aircraft trajectory classification problem also has the quality of having a more complicated 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. This input is often 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[5], 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 including Fourier descriptors[6], earth mover's distance[7], hidden Markov models[8], Hausdorff-like distances[9], Bayesian models[10] and other approaches. Most of these describe a trajectory in its entirety or compute the distance between two trajectories. We propose an alternative approach based on *trajectory features*. These features have several desirable properties. First, features based on some concise or spatially local property of a trajectory appear to correspond well to how humans envision trajectories. Second, most of the descriptors we propose can be pre-calculated *once* for each trajectory, as opposed to proximity measures such as the Hausdorff and Fréchet distances that must be computed *de novo* for every different pair of trajectories. The ability to do precomputation makes our approach suitable for 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 specifically for aircraft as well as more general work. In Section 3 we describe more carefully the specific problems we are trying to solve by designing geometric measures for aircraft trajectories. We present results and discuss the quality of the different geometric measures in Section 4. Finally, we summarize our work and offer suggestions for future work in Section 5.

1.1 Notation

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

- A trajectory $\mathbf{T}$ comprises $n+1$ timestamped points $(\mathbf{x}_0, t_0), (\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n)$, where $\mathbf{x}_i$ describes the position of point i.
- Given $\mathbf{T}$, angle θ_i is the turning angle from vector $(\mathbf{x}_i - \mathbf{x}_{i-1})$ to $(\mathbf{x}_{i+1} - \mathbf{x}_i)$. 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 $\mathbf{T}$.
- $||\mathbf{x}_n - \mathbf{x}_0||$ is the *end-to-end distance* of $\mathbf{T}$.

- $\mathcal{C}(\mathbf{T})$ is the convex hull of the points in $\mathbf{T}$. Points $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m \subset \mathbf{x}_0, \ldots, \mathbf{x}_n$ form the vertices of $\mathcal{C}(\mathbf{T})$.

- $\overline{\mathbf{A}}$ indicates the centroid of a polyline $\mathbf{A}$. Thus $\overline{\mathbf{T}}$ is the centroid of a trajectory $\mathbf{T}$ and $\overline{\mathcal{C}(\mathbf{T})}$ is the centroid of the convex hull of $\mathbf{T}$.

In the figures where flight trajectories are shown, we use color to indicate the direction of flight. Each trajectory is colored red when it starts and blue when it ends.

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 $\mathbf{T} = \{(\mathbf{x}_0, t_0), \ldots, (\mathbf{x}_n, t_n)\}$ to simply be a set of points in a $D+1$-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 distance or divergence between two trajectories. Perhaps the most straightforward measure of distance between two curves is the Hausdorff metric[11]. For two trajectories $\mathbf{A}$ and $\mathbf{B}$, the Hausdorff distance is defined as greatest distance from any point on $\mathbf{A}$ to the nearest point on $\mathbf{B}$. This gives a rough sense of the distance between two curves but neglects the direction and speed of travel along both trajectories.

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

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

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[12] 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 Fréchet 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 Fréchet distance have is that they do not allow for translational, rotational or reflectional invariance. That is, they measure the distance between two curves *given some pre-defined position and orientation*. If the curves to be compared are not already arranged as desired, they must be *aligned* before applying either the Fréchet or Hausdorff distances. This is a difficult problem. Typically one would have to do a Procrustes type of analysis to align them[13] or use an alternate method based on dynamic time warping[14] or edit distance[15] that tries to match geometric distance and curvature between points. Additionally, hidden Markov models have also been used[16] to try to compare and classify trajectories.

2.2 Why Something Different?

The measures 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 require $O(ab)$ operations to compute where a and b are the number of discrete points in the trajectories being compared. Furthermore, there is little that can be pre-computed for a trajectory in isolation: every comparison must be computed from scratch for every pair of trajectories being compared. At a more abstract level, these measures operate directly on trajectories as objects in a non-normed metric space. This makes clustering an asymptotically more difficult operation since spatial indices such as *r*-trees and *kd*-trees assume a normed vector space. Finally, the aforementioned measures each compare the entirety of two trajectories instead of identifying and addressing features 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 efficiently 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.

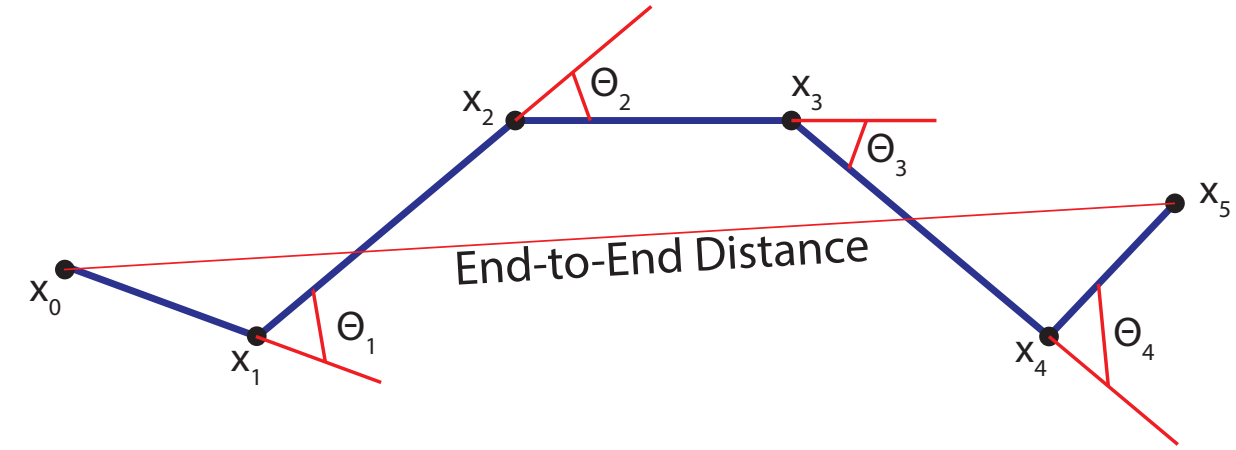
Our approach is to use simple scalar measures associated with each trajectory (such as time, total distance, etc.), and combine those values with *geometric* scalar quantities that describe the relevant geometric characteristics of the trajectory. This gives us a feature vector associated with each trajectory 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 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

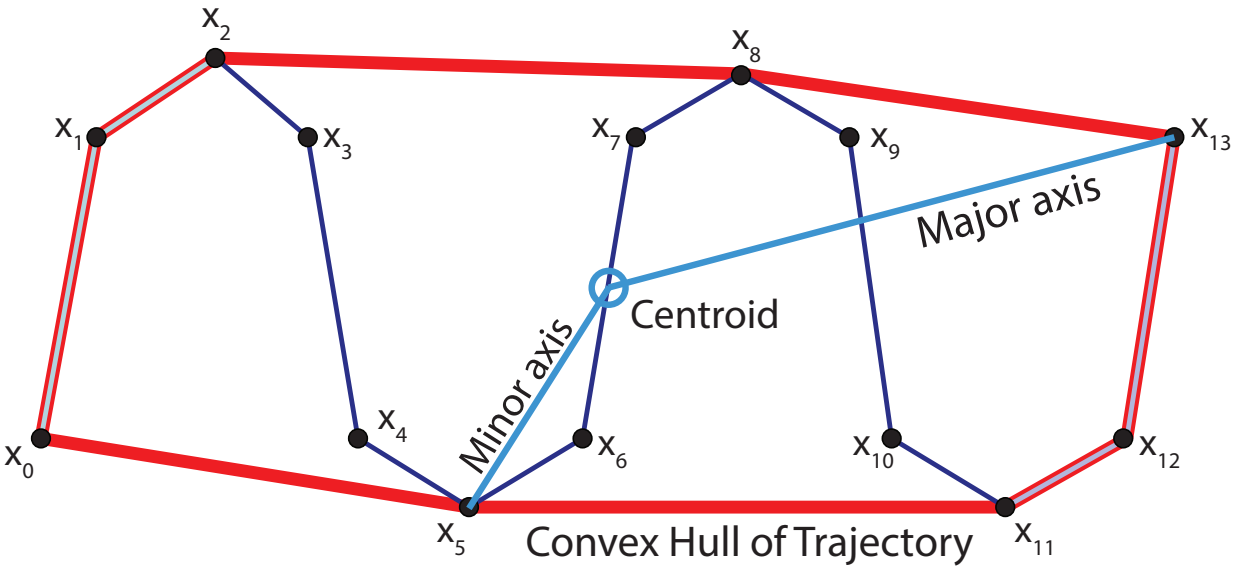
We define here more precisely what we mean by trajectory comparison. There are a few different types of problems that involve trajectory comparison. Some of the more important ones that we will cover are

- Can we find the trajectories in a database that are most similar to a given trajectory?
- Can we find trajectories that exhibit a behavior of interest without regard to translation, rotation or scale?
- Can we divide trajectories into specific clusters?
- Can we find trajectories that are outliers with respect 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.



(a)



(b)

Figure 1: Illustration of the parts and properties of a trajectory that we use to compute features. A trajectory $\mathbf{T}$ comprises $n+1$ points $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$. In (a) we see a trajectory $\mathbf{T}$ labeled with its vertices $\mathbf{x}_i$, turning angles $\theta_1 \ldots \theta_{n-1}$ and end-to-end distance $||\mathbf{x}_5 - \mathbf{x}_0||$. In (b) we see another trajectory $\mathbf{U}$ with vertices $\mathbf{x}_0 \ldots \mathbf{x}_{13}$ 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 point on $\mathcal{C}(\mathbf{U})$ and the minor axis connects the centroid with the nearest point point on $\mathcal{C}(\mathbf{U})$.

3.1 Distance Measures

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

- End-to-end distance of the flight:

$$d_e(\mathbf{T}) = ||\mathbf{x}_{n+1} - \mathbf{x}_0||$$

- Total distance traveled (length of trajectory):

$$d_t(\mathbf{T}) = \sum_{i=0}^{n-1} ||\mathbf{x}_{i+1} - \mathbf{x}_i||$$

- Distance from a given fixed point or set of points

- Centroid of points:

$$\overline{\mathbf{T}} = \frac{1}{n} \sum_{i=0}^{n} \mathbf{x}_i$$

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 defining a position (usually by longitude and latitude) and not just a single value.

3.2 Heading Measures

We 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:

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

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 Geometric 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. We approximate the length of the shortest axis as

$$\min_{\mathbf{c}\in\mathcal{C}(\mathbf{T})} ||\overline{\mathcal{C}(\mathbf{T})} - \mathbf{c}||$$

  or in words, the distance from the centroid of the convex hull to the nearest point on the convex hull. This includes *any* point on the convex hull, not just the vertices. The length of the longest axis is defined as

$$\max_{i} ||\overline{\mathcal{C}(\mathbf{T})} - x_i||$$

  for x_i on $\mathcal{C}(\mathbf{T})$. In the case of the furthest distance from the centroid, we only need to consider the vertices of the convex hull because the the convexity property of the hull.
- 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:

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

  .

- Radius of gyration of the points:

$$\sqrt{\frac{1}{n}\sum_i (\mathbf{x}_i - \overline{\mathbf{T}})^2}$$

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 this hypothesis and detailed comparisons to human studies will be left to future work.

We will also use one final geometric measure based on the concept of *distance geometry*[17] to describe complex shapes in more detail. We define it as follows. First, parameterize a trajectory uniformly over the interval $t \in [0, 1]$. Then choose a set of m intervals (t_{m_1}, t_{m_2}) and measure the distance between the corresponding points $\mathbf{x}_{m_1}$ and $\mathbf{x}_{m_2}$. 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. Further, if we normalize these m values by the largest value so that that all of the values are between 0 and 1, we obtain a measure that is also *scale invariant*. The distance geometry approach implicitly allows for features that are not neccesarily global in nature if the intratrajectory distances include smaller ranges that describe distances in smaller parts of the tracks. This helps to capture abnormal local aspects of the trajectory that have shown to be important in [18] and [19].

3.4 Use of Feature Vectors

The feature vector representation enables two different approaches to solve the problems listed above. The first is the most straightforward. We can calculate the feature vectors and then use traditional searching 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 we choose the feature vector carefully and build a distance metric on those vectors that is expressible as an L^p norm, then we can use a spatial indexing scheme such as an *R-tree*[20] to store feature vector values, search for nearest neighbors, and even do clustering. Many clustering algorithms allow for outliers to be specifically identified, which gives a powerful method of defining and determining anomalous flights.

4 Results

4.1 Trajectory Data Set

We tested our algorithms and generated the results shown in this paper using the ASDI (Aircraft Situation Display to Industry) data set. This is an air traffic data set generated by the US FAA (Federal Aviation Administration) that contains most US civilian flights that have flight plans on file. We obtain the data via a subscription through AirNav, LLC, which disseminates the traffic data in XML format along with additional metadata concerning each flight.

The ASDI data set comprises approximately 50,000 flights per day. At present we have approximately 6 months of archived data. Each flight consists of a sequence of data points generally spaced 60 seconds apart. Each data point contains a flight ID, a timestamp, position data (latitude, longitude, heading) and a large amount of supporting metadata. Flights contain anywhere between ten and several hundred data points. Although the majority of the data points in most flights are uniformly spaced every 60 seconds, the data contains occasional dropouts and duplicates depending on contact between an aircraft and the ground sensors that communicate with it.

Metadata in the ASDI data set often includes altitude, speed, departure/arrival airport, departure/arrival times and so on. This can be very useful in classifying flights. However, the focus of this work is to study how geometric features can be used to compare, contrast and identify flights. We do not use any of the metadata for this task.

4.2 Data Cleaning and Trajectory Assembly

The data points in the ASDI feed arrive sorted by timestamp rather than by flight. Our first task was to reorganize this stream into potential trajectories. We first sort by flight ID to create streams belonging to different flight IDs then search each stream for large time breaks between points that indicate multiple stops under a common flight ID. We used a threshold value of 30 minutes to identify these breaks. Values between 20 and 60 minutes did not yield significantly different results.

Once we assembled these candidate trajectories we ran each one through a simple clean-

ing operation to remove obviously bad data. We looked for and removed data points that were an unreasonably large distance away from their neighbors given the time separation between them. In this case, "unreasonably large distance" required an airspeed 3-10 times faster than a typical airplane. This was sufficient to remove the especially bad points. There are certainly more sophisticated cleaning and filtering operations available. We chose not to use them because we want to test our measures for robustness against data that may contain significant uncertainty or noise in the position fields.

4.3 Simple Geometric Filtering

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

4.3.1 Avoiding Airspace

One possible question that we could ask regarding a collection of flights is, "Is there a section of airspace that flights seem to avoid?" 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 flying circles as part of training), but traveled a distance that was significantly 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 aspect ratio of the convex hull, requiring the flights to be more "round". These criteria turned up a sizable cluster of flights on July 10, 2013 shown in Figure 2. Upon further research we found out what the flights were avoiding. That day, many flights were rerouted to avoid a large system of thunderstorms that swept eastward through Illinois and Indiana all the way to Ohio and Pennsylvania. In Figure 3 we display the "avoiding" trajectories again along with a weather map from that day.

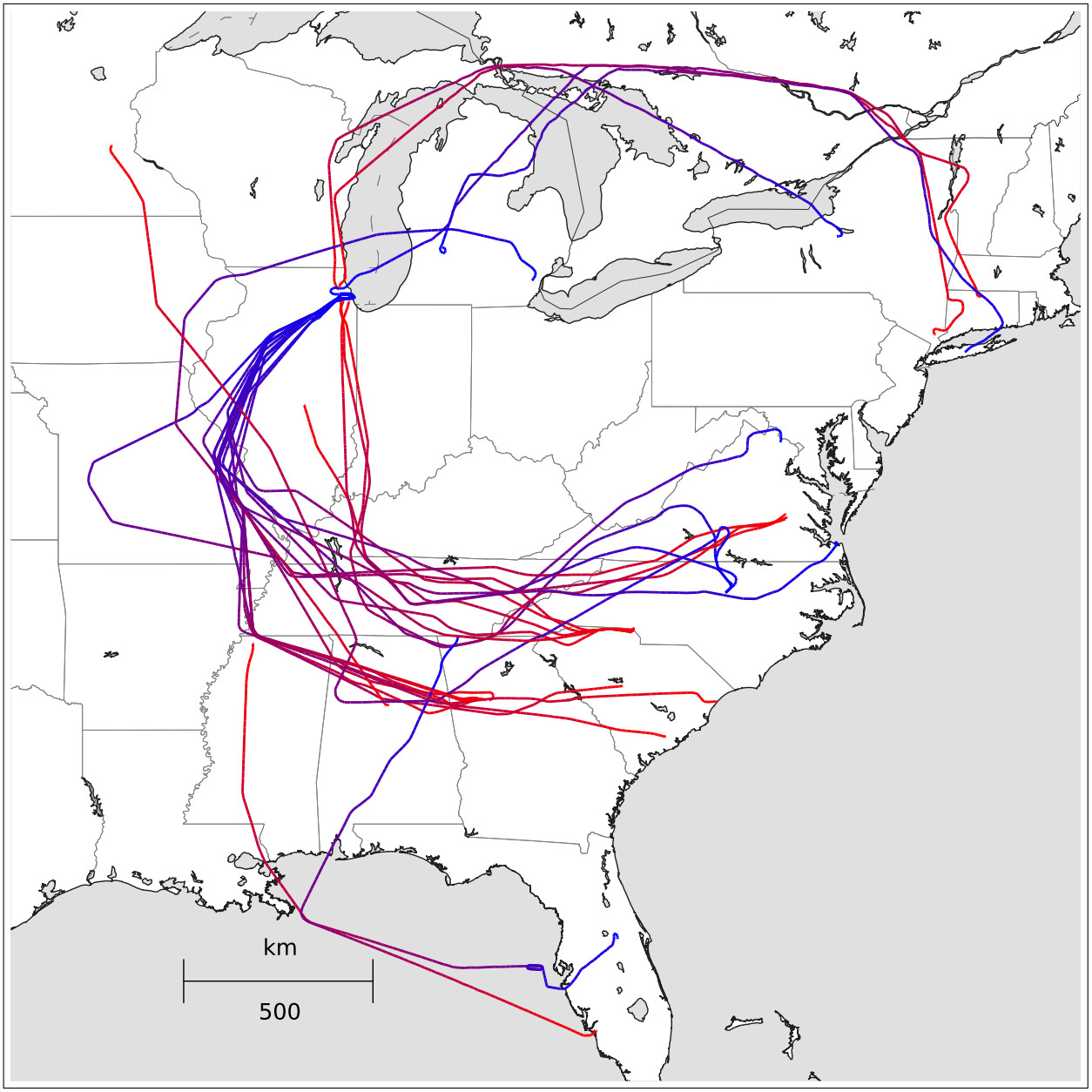


Figure 2: Examples of flights found for the "avoiding" specification. In this case, we required the end points of the flight to be at least 1000 kilometers apart, the ratio of the end-to-end distance of the flight to the total flight distance to be less than 0.7, and the aspect ratio of the convex hull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

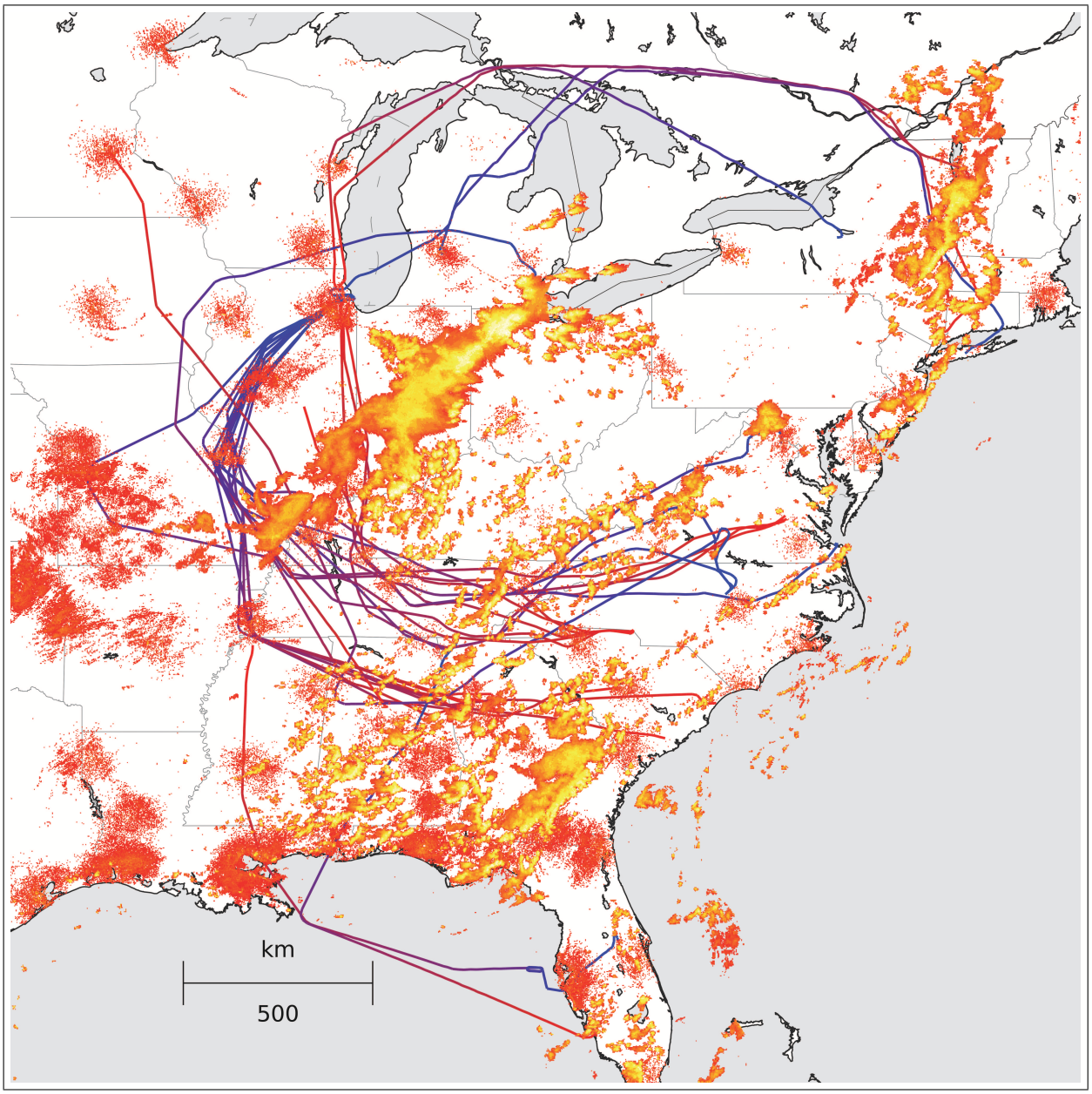


Figure 3: Most of the trajectories identified with the "avoiding" specification were responding to this event: a severe weather system crossing most of the midwestern United States from Illinois to New York. This weather map was captured at 8:30PM Eastern Daylight Time (UTC-5) on July 10, 2014.

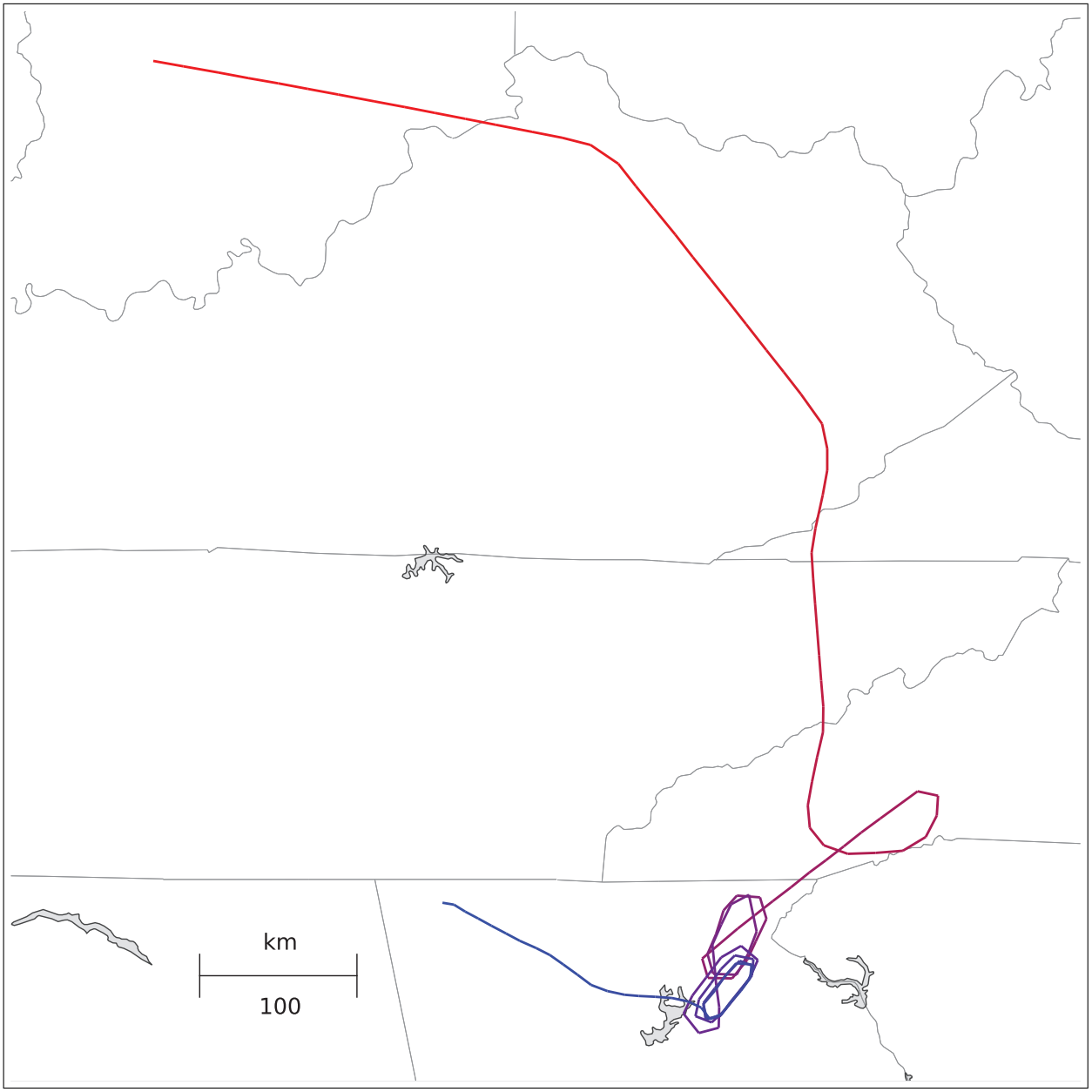


Figure 4: Examples of a flight found for the "holding and diverted" specification. In this case, we required the end points of the flight to be at least 200 kilometers apart, the total amount of turning to be at least 20π radians, and the aspect ratio of the convex hull to be at least $\frac{1}{10}$.

4.3.2 Holding Pattern

Another distinctive pattern of interest in flight trajectories is a holding pattern. We define this as a flight that flies for some distance and then enters a circling pattern due to some sort of landing delay. We translated this into two geometric constraints. First, the flight had to have at least moderate length (200km) and a significant total curvature that would be unusual for a point-to-point flight (at least 20π).

This search returned many flights that had clearly been instructed to circle while awaiting permission to land. We decided to extend it for a more difficult test of our approach to search for flights that entered holding patterns and were ultimately diverted to different airports. To accomplish this, we added the constraint that the aspect ratio of the convex hull of the trajectory must have an aspect ratio of at least 0.1. This search turned up one flight in our test data set (see Figure 4). When we examined the original metadata for this flight we found that it was indeed inbound to Atlanta when it entered a holding pattern and was finally diverted to Chattanooga in early June of 2013.

4.3.3 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 scanning, or boustrophedon, pattern. This type of flight will have a significant length, but will be enclosed by a fairly compact shape. For this search, we require a reasonably long total distance, but a small radius of gyration. An example of these flights is shown in Figure 5. 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 searched for straight segments separated by 180 degree turns that also did well, but was brittle with respect to minor variations in the mapping process.

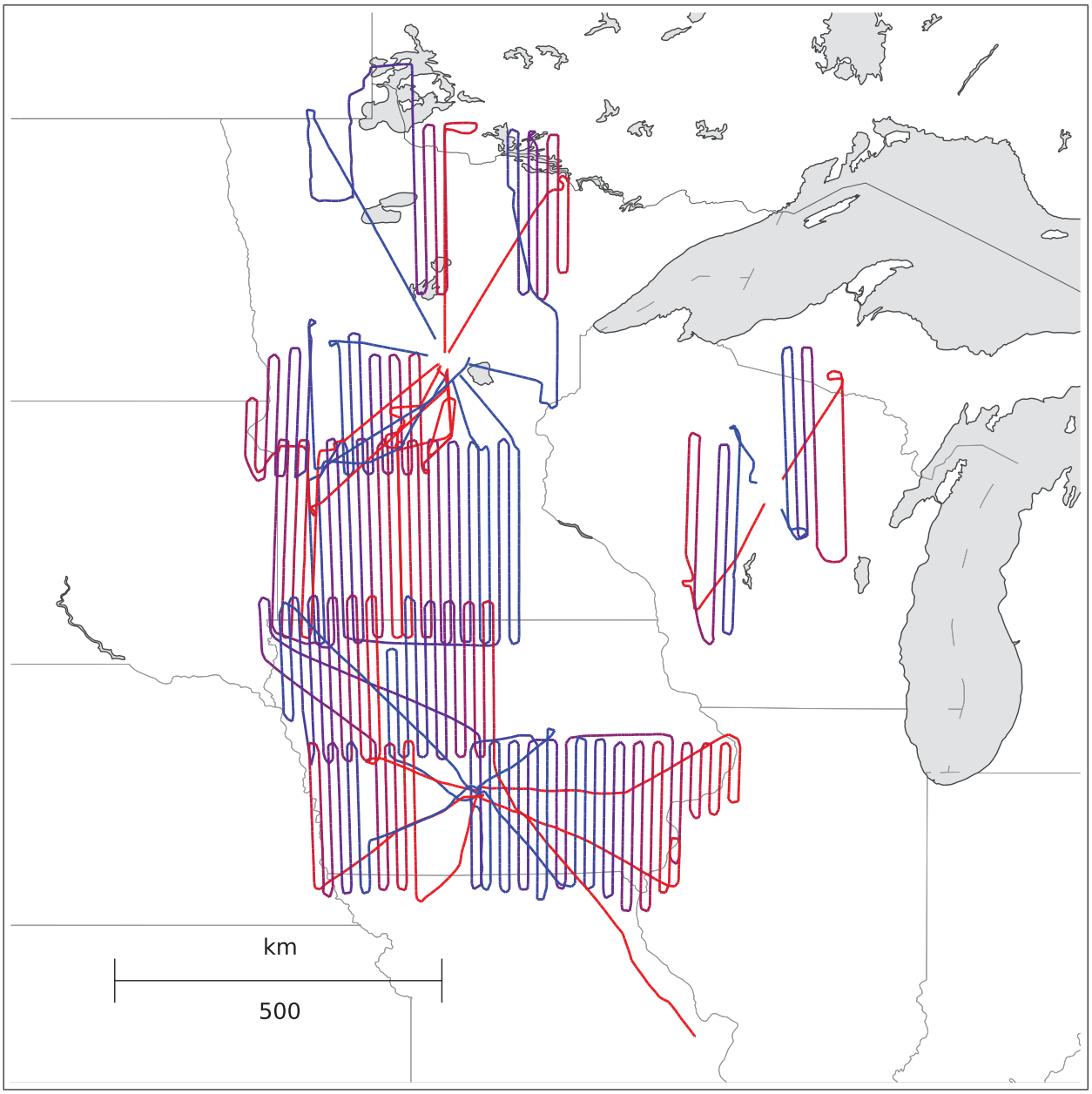
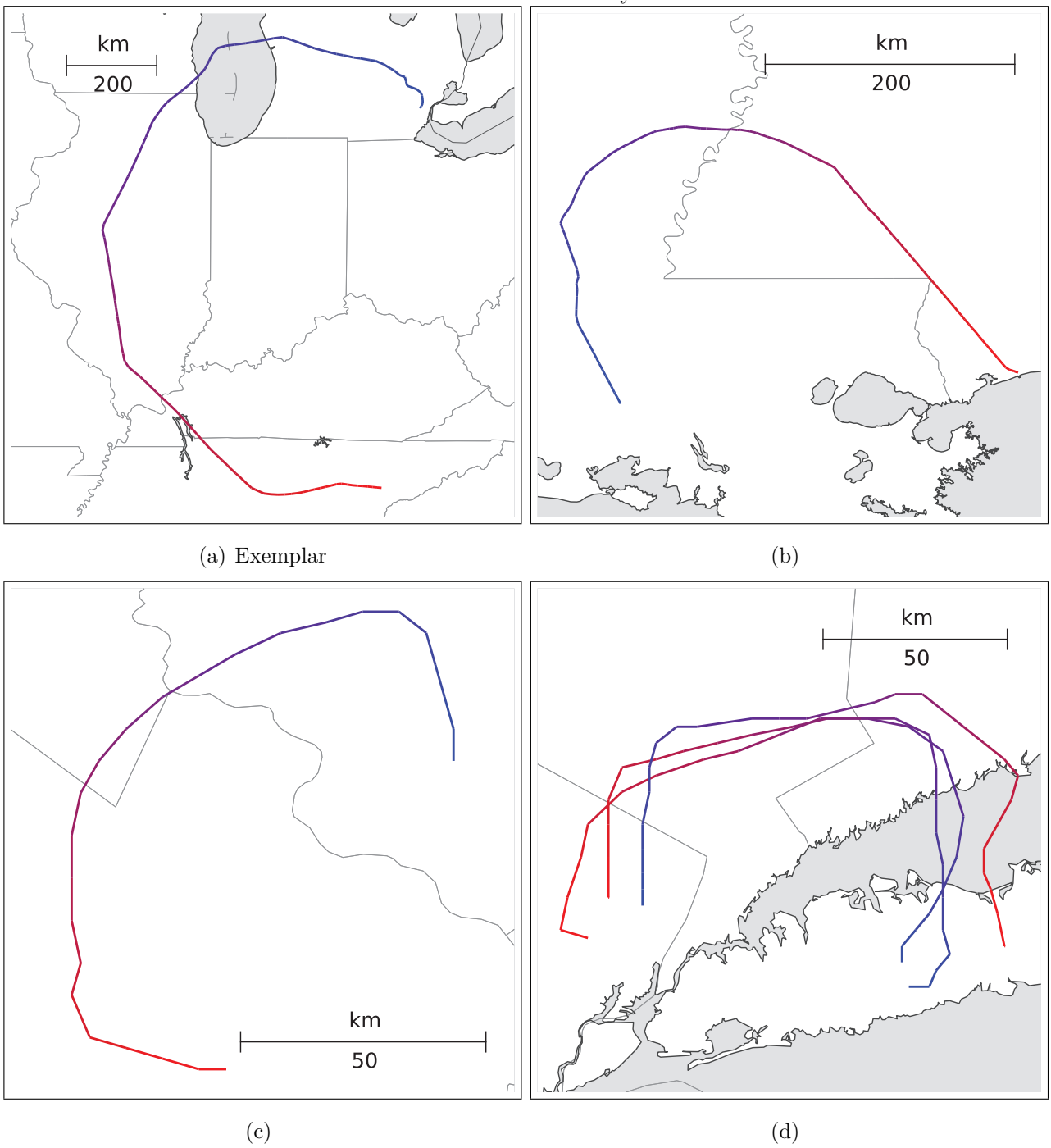


Figure 5: Examples of flights found for the mapping criteria. We require the flights to be longer than 800 miles, and have a convex hull aspect ratio greater than 1/8. We then took the top 50 flights in terms of having a small ratio of radius of gyration to total distance flown. A small sample of the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



(a) Exemplar (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), although at very different scales than (a). The curve in (b) flies around the southern Louisiana area, while the curve in (c) flies around Washington, DC. Finally, in (d), we see examples of flights diverting around New York City, in both directions.

4.4 Distance Geometry Examples

To demonstrate the distance geometry technique described in Section 3, we will use one of the flights that was found above in the avoiding airspace example above (Figure 2). While the goal in that example was achieved by describing the features, distance geometry enables an even simpler solution. We begin with the flight shown in Fig 6(a), measure distances at various points along the flight, and build a feature vector with the distances normalized to fall between 0 and 1. This gives us a feature vector based solely on the relative distances between different points in the trajectory. We then compare this feature vector to those from other flights in the database using the L^2 norm to find flights with a similar shape.

In our example, we chose 10 different distances to use as the intratrajectory distances. Let $\mathbf{T}(t)(t \in [0,1])$ be the entire trajectory parameterized by t. Let $d : \mathbb{R}^2 \times \mathbb{R}^2 -> \mathbb{R}$ be a distance function between points (here the familiar Euclidean distance). We then define the following distances as our features:

- End-to-end distance: $d(\mathbf{T}(0), \mathbf{T}(1))$
- Distances from midpoint to beginning and end: $d(\mathbf{T}(0), \mathbf{T}(\frac{1}{2}))$ and $d(\mathbf{T}(\frac{1}{2}), \mathbf{T}(1))$
- Thirds: $d(\mathbf{T}(0), \mathbf{T}(\frac{1}{3}))$, $d(\mathbf{T}(\frac{1}{3}), \mathbf{T}(\frac{2}{3}))$, $d(\mathbf{T}(\frac{2}{3}), \mathbf{T}(1))$
- Quarters: $d(\mathbf{T}(0), \mathbf{T}(\frac{1}{4}))$, $d(\mathbf{T}(\frac{1}{4}), \mathbf{T}(\frac{1}{2}))$, $d(\mathbf{T}(\frac{1}{2}), \mathbf{T}(\frac{3}{4}))$, $d(\mathbf{T}(\frac{3}{4}), \mathbf{T}(1))$

While we could have estimated the distances at the precise time points through interpolation between the nearest discrete points, we simply chose the points 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 compared to precise interpolation between points.

The results for that the distance geometry search are shown in Figure 6. There were a wide variety of results, all with similar fundamental shapes but with a wide variety of sizes and orientations. We had also originally attempted these comparisons with curve alignment algorithms that were based on dynamic programming techniques. Those approaches took much longer due to their increased computational complexity and failed to match the global shape of the curves due to their focus on aligning local structures.

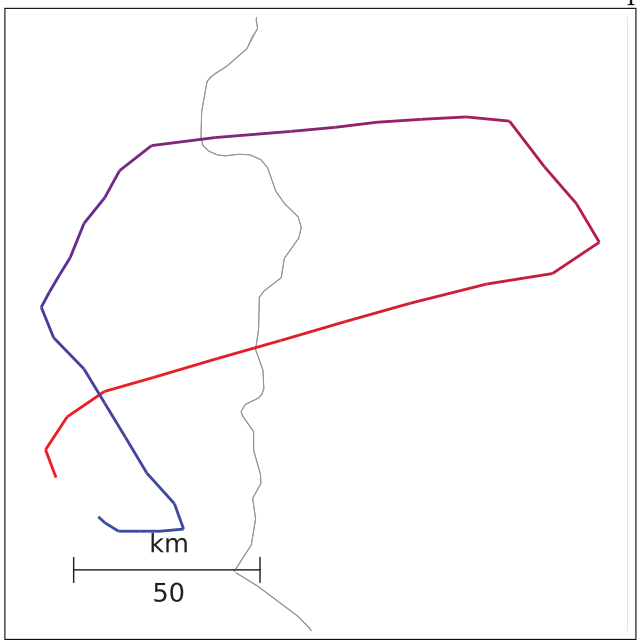
4.5 Indexing Within Feature Space

We have just demonstrated finding trajectories with a certain shape by calculating a feature vector for an exemplar and then comparing that exemplar to all trajectories in the database one by one - an $O(n)$ search with respect to the size of the database. This becomes expensive as the database grows to millions or billions of trajectories, especially since each search has to be computed from scratch. An approach that would allow us to re-use calculations is to create a spatial index within multidimensional feature space that will allow us to search quickly for nearby flights.

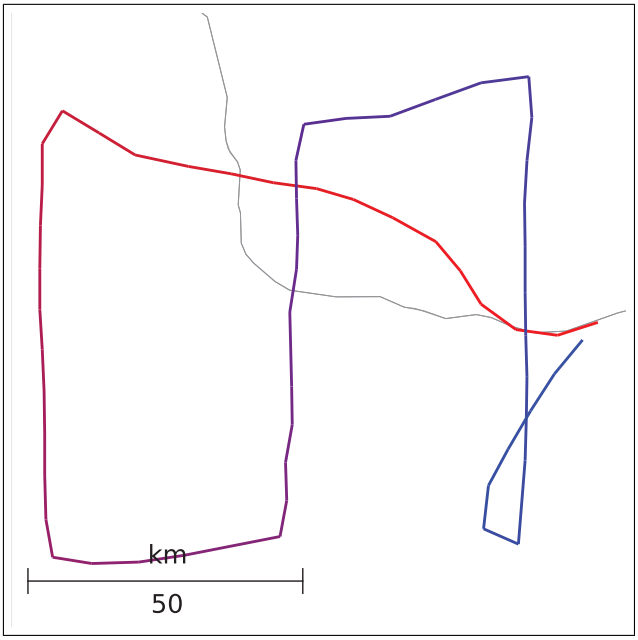
One of the most popular data structures for this type of spatial indexing is an *R-tree*[20]. The R-tree is a multidimensional data structure that represents objects by their minimum bounding n-dimensional rectangle in the next highest level of a tree. This hierarchical structure allows for logarithmic 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), a roughly figure-eight shape which is somewhat unusual among the flights in our database. It is more difficult to write a feature descriptor for this flight. Instead of writing the descriptor directly, we define the different dimensions of the feature space to be features that we guess will be relevant. For this test we chose three 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 built an index over approximately 50,000 flights (about 1 day's worth) and asked for the 10 closest points in feature space. The flights corresponding to three of the closest points are shown. Given the small dimension of the feature space, some of the other neighbors did not resemble the figure-eight shape as closely. On an interesting note, we 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.

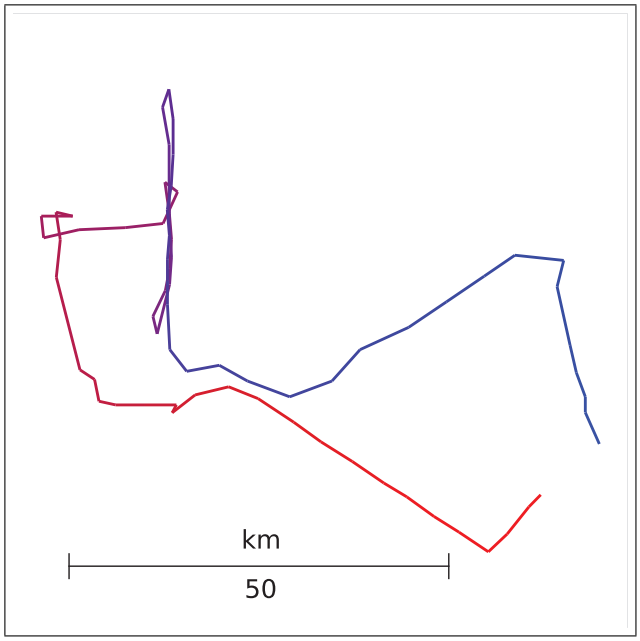
Representing the data as feature vectors in a normed vector space also allows clustering to be done in a number of different ways. There are a variety of traditional dimensionality



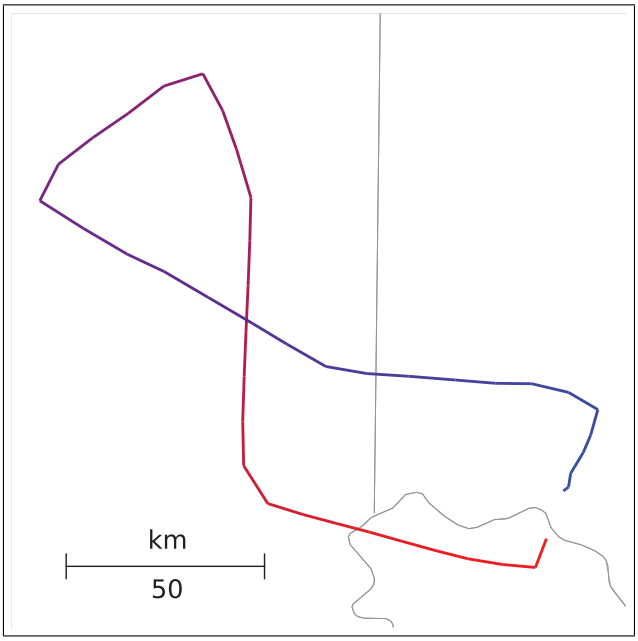
(a) Exemplar



(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.

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, the feature space embedding enables an 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.

5 Conclusions

For many cases, working in feature space rather than the physical space in which trajectories are embedded is a more effective way of finding trajectories that match a given set of criteria than using dynamic programming approaches that employ more local comparisons. This is partially due to computational issues, but very preliminary discussions 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 tools for analysts to use to find trajectories that correspond to specific *behaviors* and not necessarily to narrowly-defined numerical quantities.

We anticipate that follow-on work will focus on two general areas. The first will center 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 work with analysts to understand better how people currently compare trajectories based on their experience.

6 Acknowledgments

We would like to acknowledge Randy Brost, Hyrum Anderson, Eddie Ochoa and Cindy Phillips for useful initial discussions regarding the problem of classifying trajectories. 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.

References

[1] Defense Industry Daily. Too much information: Taming the UAV data explosion. *Defense Industry Daily*, May 16 2010.

[2] Yu Zheng and Xiaofang Zhou, editors. *Computing with Spatial Trajectories*. Springer, 2011.

[3] Toby A. Patterson, Len Thomas, Chris Wilcox, Otso Ovaskainen, and Jason Matthiopoulos. State space models of individual animal movement. *Trends in Ecology & Evolution*, 23(2):87 – 94, 2008.

[4] Alexander V. Popov, Yury N. Vorobjev, and Dmitry O. Zharkov. MDTRA: A molecular dynamics trajectory analyzer with a graphical user interface. *Journal of Computational Chemistry*, 34(4):319–325, 2013.

[5] Zhouyu Fu, Weiming Hu, and Tieniu Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–602–5, Sept 2005.

[6] R. Annoni and C.H.Q. 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.

[7] Francesca Boem, Felice Andrea Pellegrino, Gianfranco Fenu, and Thomas Parisini. Multi-feature trajectory clustering using earth mover's distance. In *CASE*, pages 310–315. IEEE, 2011.

[8] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfeld. Object trajectory-based activity classification and recognition using hidden Markov models. *IEEE Trans. Image Process*, 2005, 2007.

[9] Bo Guan, Liangxu Liu, and Jinyang Chen. Using relative distance and Hausdorff distance to mine trajectory clusters. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(1):115–122, 2013.

[10] Julian F. P. Kooij, Gwenn Englebienne, and Dariu 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.

[11] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer Verlag, 1998.

[12] Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Technische Universitat Wien, April 1994.

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

[14] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *KDD Workshop*, pages 359–370. AAAI Press, 1994.

[15] Lei Chen, M. Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 491–502, New York, NY, USA, 2005. ACM.

[16] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic. Discovering clusters in motion time-series data. In *Proc. Computer Vision and Pattern Recognition (CVPR '03)*, pages 375–381, 2003.

[17] G.M. Crippen and T.F. Havel. *Distance geometry and molecular conformation*. Chemometrics series. Research Studies Press, 1988.

[18] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 593–604, New York, NY, USA, 2007. ACM.

[19] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. Trajectory outlier detection: A partition-and-detect framework. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 140–149, Washington, DC, USA, 2008. IEEE Computer Society.

[20] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, pages 47–57, New York, NY, USA, 1984. ACM.