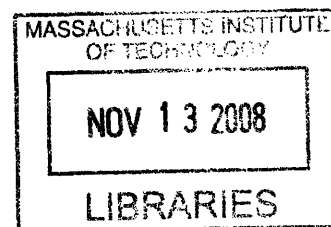


Automated Identification of Terminal Area Air Traffic Flows and Weather Related Deviations

by
Tony Ng

S.B. EECS, M.I.T., 2007



Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 2008

©2008 Massachusetts Institute of Technology. All rights reserved.

This work was sponsored by the Air Force under Air Force Contract
FA8721-05-C-0002. Opinions, interpretations, conclusions, and
recommendations are those of the authors and are not necessarily endorsed
by the United States Government.

Author _____
Department of Electrical Engineering and Computer Science
May 23, 2008

Certified by _____
Richard DeLaura
Technical Staff, M.I.T. Lincoln Laboratory
VI-A Company Thesis Supervisor

Certified by _____
Patrick H. Winston
Professor of Computer Science
M.I.T. Thesis Supervisor

Accepted by _____
Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

ARCHIVES

**Automated Identification of Terminal Area Air Traffic Flows
and Weather Related Flight Deviations**

by
Tony Ng

Submitted to the
Department of Electrical Engineering and Computer Science

May 23, 2008

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Air traffic in terminal air space is very complex, making it very difficult to identify air traffic flows. Finding air traffic flows and flow boundaries are very helpful in analyzing how air traffic would react to weather. This thesis created the Terminal Traffic Flow Identifier algorithm to solve this problem. The algorithm was demonstrated to work in the Atlanta by quickly processing 20,000 sample trajectories and returning accurate flows with tight boundaries. This thesis also created techniques to extract weather features that occur inside the identified flows and demonstrated that training upon these features give good results. The algorithms and software created in this thesis may soon be incorporated into larger traffic managements systems developed at MIT Lincoln Laboratory.

VI-A Company Thesis Supervisor: Richard DeLaura
Title: Technical Staff, M.I.T. Lincoln Laboratory

M.I.T. Thesis Supervisor: Patrick H. Winston
Title: Professor of Computer Science

Acknowledgements

My supervisor Richard DeLaura at MIT Lincoln Laboratory was extremely helpful. I would like to thank him for hiring me for multiple internships at Lincoln Laboratory and giving me the opportunity to work on this thesis. He gave me a lot of freedom to investigate solutions for myself and was always available to provide guidance if needed.

Brian Martin was very helpful in showing me how some of the existing tools worked. He helped me figure out how to extract weather data from CIWS Cartesian files and gave me various scripts that helped with plotting weather. Without him, it would have taken me a lot longer to figure out how the existing systems at Lincoln Laboratory worked.

Thanks to Professor Patrick Winston for being my thesis advisor. His artificial intelligence class gave me the background I needed to do this thesis and his advice on my writing was helpful.

Thanks to Bill Moser for steering me away from using a network node model for terminal airspace, which would have made things a lot more difficult and less accurate. Thanks to Amar Boghani for helping me setup the SQL database that stores processed ETMS data. Thanks to my officemate Christina Leber for listening to me complain about the problems with my thesis.

Finally, I thank my family. None of this would be possible without their love and support.

Table of Contents

1. Vision	8
2. Steps	10
2.1 Data Sources	10
2.1.1 Enhanced Traffic Management System (ETMS)	11
2.1.2 Corridor Integrated Weather System (CIWS)	11
2.2 Air Traffic Flows	12
2.3 Flow Boundaries	14
2.4 Flight Responses to Weather Conditions	15
2.4.1 Vertically Integrated Liquid (VIL)	15
2.4.2 Echo Tops	18
2.4.3 Widest Traversable Gap	20
3 News	21
3.1 Modeling the Terminal Airspace	21
3.2 Clustering Algorithms to Identify Flows	26
3.2.1 K-Means	29
3.2.2 Hierarchical Clustering	32
3.2.3 Hybrid K-Means Hierarchical	33
3.3 Estimating Flow Boundaries	36
3.4 Widest Traversable Gap Algorithm	39
3.5 Application and Analysis	42
4. Contributions	46
5. References	47

List of Figures

2.1	Departing Flights in Atlanta	13
2.2	VIL Plot of Thunderstorm	17
2.3	Echo Top Plot of Thunderstorm	19
3.1	Circular Coordinate Model for Terminal Airspace	24
3.2	Special Case: Trajectory Contains Sharp Turns	25
3.3	Special Case: Spiral-like Trajectories	26
3.4	Set of Trajectories Represented as a Matrix	27
3.5	Matrix of Trajectories Being Clustered	28
3.6	Pseudo-code for K-Means	30
3.7	Agglomerative Hierarchical Clustering	32
3.8	Hybrid K-Means/Hierarchical Clustering	34
3.9	Flights Crossing Circle Around Airport	37
3.10	Histogram of Intersections of a Cluster of Trajectories with Circle	38
3.11	Flow Projected onto Matrix with Weather	39
3.12	Matrix of Traversable Regions	40
3.13	Reduced Matrix of Traversable Regions	41
3.14	Trajectories and Flow Boundaries	43
3.15	Flows, VIL, and Echo Tops during a Thunderstorm	44

1. Vision

Every day, air traffic controllers handle tens of thousands of flights in the United States. These flights include commercial, private, military, and cargo planes. At any given moment, there are several thousand planes flying above the United States. In order to keep flight traffic organized, most planes follow preplanned routes and schedules. Unfortunately, nature is not always permitting. When harsh weather and storms disrupt these plans, air traffic controllers must improvise and create new traffic routes.

Successful management of air traffic during harsh weather conditions is very difficult. Not only do airplanes need to be assigned new routes that allow them to safely navigate through weather, delays need to be minimized. People who have experienced weather delays and waited at airports know how annoying it can be. Delays are costly to both the customers and the airlines.

One part of air traffic management that has not been well researched is how planes move in terminal area airspace. The terminal area airspace is very complex because planes are constantly taking off and landing at a very tight space. A better understanding of how planes move in the terminal area would allow for a better, more accurate weather analysis. Predictive models of how air traffic will react to weather allow air traffic controllers to perform their jobs better, which would help ensure safety and prevent delays.

MIT Lincoln Laboratory contributes to the research and development of Federal Aviation Administration (FAA) weather systems and the integration of weather data into air traffic management. I was given the opportunity to work with the laboratory to create the Terminal Traffic Flow Identifier, a software algorithms package that would automatically examine a large collection of data to identify air traffic flows and extract features that can

be used to perform a weather analysis. This software package may be integrated into future air traffic management systems that will contribute to the vision of making air traffic safer and more efficient.

One likely application of the software created by this thesis is to predict traffic deviations based on past data. If planes are capable of penetrating specific weather conditions in the past, it is likely that planes will be able to penetrate similar conditions in the future.

Traffic flows and boundaries, which determine whether a plane is deviating or not, can be determined by using the Terminal Traffic Flow Identifier on existing data. Features from flight trajectories during severe weather can be extracted for cases where planes successfully penetrated weather and for cases where planes were forced to deviate. These sample trajectories and features can be used to create a classifier that would predict whether or not planes will deviate during future weather conditions. Such a predictor would allow an air traffic controller to look at a weather forecast and be able to determine which traffic flows would be shut down and which ones would still be operational.

2. Steps

In order to figure out how weather affects traffic flows in terminal air space, there are several tasks that need to be accomplished. The following is an outline of the steps involved:

1. Find a source of flight trajectory and weather data.
2. Find algorithm to automatically identify air traffic flows in clear weather.
3. Find algorithm to automatically estimate boundaries of air traffic flows.
4. Find a way to analyze weather conditions in flows to find out what causes flights to deviate.

This thesis focuses on the Hartsfield International Airport in Atlanta (ATL). Air traffic near Atlanta is well organized, making the problem much easier. This thesis attempts to demonstrate a working technique rather than methods to deal with subtleties in more complex airspaces. The programs in this thesis are written in MATLAB.

2.1 Data Sources

MIT Lincoln Laboratory has access to data sources from several air traffic management systems that are used by the FAA. Two of these systems are the Enhanced Traffic Management System (ETMS) and the Corridor Integrated Weather System (CIWS). Since this thesis is being done for MIT Lincoln Laboratory, access to select archives from these two systems was available.

2.1.1 Enhanced Traffic Management System (ETMS)

ETMS is a real time aircraft tracking system used by the FAA and air traffic management personnel to direct aircraft flow in the United States. ETMS provides flight data on nearly all flights in the United States. Every minute, ETMS records the time, altitude, latitude, longitude, heading, and speed of flights. It also provides the origin airport, destination airport, and flight plans. Flight plans are a list of fixes, or waypoints, that a plane attempts to travel through in order to reach its destination.

ETMS archives contain daily flight data for the last several years. This amount of data is unwieldy and difficult to sort through. Therefore, days of interest are identified by looking at weather histories. ETMS data for these days are loaded onto an SQL database for easy access.

2.1.2 Corridor Integrated Weather System (CIWS)

Weather data can be obtained from CWIS [1]. The system is designed to support efficient and safe management of air traffic during thunderstorms by providing very accurate, low latency, high resolution three dimensional weather information and forecasts. CIWS provides information about intensity, spatial distribution, vertical structure, and the evolution and morphology of convective weather and thunderstorms. CWIS is updated every five minutes and has a spatial resolution of one kilometer.

The main products of interest from CWIS are precipitation intensity, expressed as vertically integrated liquid (VIL), and echo tops. CWIS can provide VIL and echo top data at five minute intervals throughout the day. It also provides VIL and echo top forecasts, also at five minute intervals, for up to two hours into the future. However, this thesis will not attempt to incorporate weather forecasts, since weather forecast errors translate into errors in the identification of the weather that pilots will choose to penetrate or avoid. Once the pilot weather avoidance behavior is understood for observed weather,

predictions of pilot avoidance may be based on weather forecasts and the source of prediction errors, weather forecasts or pilot variations, may be more readily determined.

2.2 Air Traffic Flows

Air traffic tends to be very dense in certain regions of airspace, particularly in between airports and fixes. Paths that are frequented heavily by airplanes are air traffic flows. A traffic flow consists of a path that planes are trying to follow and boundaries around the path that planes are likely to remain inside as it follows the path. Air traffic flows are important for creating models of airspace to be used to predict the effects of weather and to find efficient ways to reroute traffic.

Air traffic tends to be extremely dense near airports because all flights must start and terminate at an airport. This area is known as the terminal airspace. Terminal airspace differs from en route airspace, which is further away from airports, because air traffic controllers have a much higher degree of flexibility in routing planes. In en route airspace, planes are usually traveling at a constant cruising speed and altitude. Also, en route airspace is very well structured and planes tend to take beeline paths from fix to fix, making flows in en route airspace easy to define. However, in terminal airspace, planes are either taking off or landing. Some variables such as the type of plane and the weight of cargo affect how quickly a plane can ascend or descend and its turning ability. Planes are also limited in how they can avoid weather or other obstacles near airports. If a plane is flying near an airport, it is usually taking off or landing. The altitude is fairly limited because the plane must descend into the airport or ascend away from the airport at a safe comfortable speed. For example, a plane cannot fly over a thunderstorm while landing at the same time, because landing requires the plane to descend. A combination of all these factors makes the terminal airspace very complex and very different from en route airspace. Terminal airspace is less restrictive and less defined in order to allow planes to make the right adjustments to maneuver through the terminal area. Therefore there are no well defined flow boundaries for air traffic managers to work with.

One way to accurately identify these flows is through a statistical approach. Many similar trajectories can be found when looking at a large number of flights over several different clear weather days. These similar trajectories are members of the same traffic flow. By gathering enough data and having enough sample trajectories for each flow, characteristics of the flow can be estimated from these samples. Figuring out a good way to automatically identify these flows from a given set of data is the goal for this step of the project.

Departing Flights in Atlanta

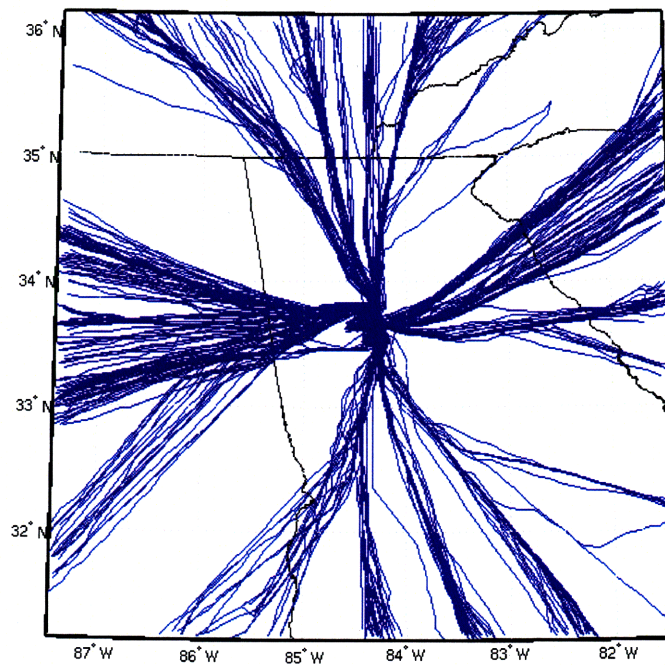


Figure 2.1. All departing flights in Atlanta over the course of one day. Looking at all the trajectories, it is obvious that there is some structure to it and there are regions containing large concentrations of trajectories. These concentrations are traffic flows.

2.3 Flow Boundaries

In order for traffic flows to be useful, each flow needs to have flow boundaries because the boundaries indicate whether a plane is deviating or not. The traffic flow boundaries define the area that planes will likely remain inside if it is following that specific traffic flow. Planes tend to leave the flow boundaries only if there is an obstruction, such as hazardous weather conditions.

Air traffic controllers must ensure a safe separation between airplanes at all times. When a pilot refuses to go through weather, he will ask air traffic control for permission to deviate to a new route. Air traffic control may grant the request or suggest another route. Whatever happens, the air traffic controller is responsible for ensuring that separation standards are not violated by airplanes flying too close to neighbors. If too many pilots are requesting deviations, air traffic control may choose to shut down a route or traffic flow because it is impossible to accommodate the deviations that pilots request and keep a safe separation between planes.

Having accurate boundaries for air traffic flows is important because it determines how much separation planes require. If the boundaries are too large, too much airspace would be reserved and air traffic controllers would lose some of the freedom in rerouting air traffic. If the boundaries are too small, air traffic controllers may accidentally route air traffic too close together, which would increase the chance of accidents occurring.

A statistical approach can be taken to figure out flow boundaries. Assuming enough flight trajectories can be gathered for a particular flow, all those flows can be analyzed and a boundary can be created from that data. Finding a way to automatically find the boundaries of traffic flows is the goal for this step of the project.

2.4 Flight Responses to Weather Conditions

In order to figure out how flight trajectories change in response to weather conditions, measurements for weather are needed. The two most commonly used predictors for severe weather conditions are VIL and echo tops which, as mentioned earlier, is provided by CIWS. Individually, each of these measurements is strongly correlated with severe weather. Combined, the VIL and echo tops complement each other well and accurately indicate severe weather conditions. Naturally, the more severe the weather conditions in a traffic flow, the more likely that planes traveling in that flow will deviate.

Another approach to figuring out what happens to flight trajectories in response to weather conditions is to examine the options available to the pilot or air traffic controller. Basically, what is the best path to avoid weather while staying within the traffic flow boundaries? If this path is sufficient, then the pilot will probably take it and remain inside the flow. If the best possible path is not safe enough, then the plane will be forced to deviate.

By looking at the severity of weather conditions inside a traffic flow and examining the possible options to fly through the weather, it should be possible to make a decent guess at whether a plane will deviate out of the flow or not. The goal of this step of the project is to analyze a large number of storm cases and figure out a reasonably accurate way to guess whether a plane will deviate outside of its flow boundaries or not.

2.4.1 Vertically Integrated Liquid (VIL)

VIL is an estimate of the amount of precipitation in the clouds. More precisely it is the vertical integral of liquid water content over a column of air. The amount of water content is estimated by reflectivity, which is measured by weather radar observations at different elevation angles. More precisely, water content is computed from the equivalent reflectivity factor Z_e and is defined as

$$M = 3.44 \times 10^{-3} Z_e^{4/7}$$

where the units of M are in grams per cubic meter [2]. Z_e depends on the measured reflectivity of the target and the radar wavelength. VIL typically measured in kilograms per square meter.

A larger VIL means that there is more precipitation in the column of air. VIL is correlated with hail size, but VIL alone cannot accurately determine the severity of hail because it also depends on some other variables. However, for larger VIL values, it is very strongly correlated with severe thunderstorms that airplanes cannot fly through, so it is very useful in this respect.

VIL is different from the typical weather radar images, such as those seen in weather forecasts on TV. Those weather images show reflectivity, which is a measure of radar return power, measured in decibels. VIL is used in air traffic management because it is believed that VIL predicts the conditions that pilots tend to avoid better than reflectivity [3]. VIL data is usually examined visually and it is standard to map VIL intensities onto a six color scale known as the Video Integrated Processor scale [4]. VIL intensities are often referred to by this scale (“level one”, “level six”, etc). Level three precipitation is generally considered to be “intense”.

VIL Plot of Thunderstorm

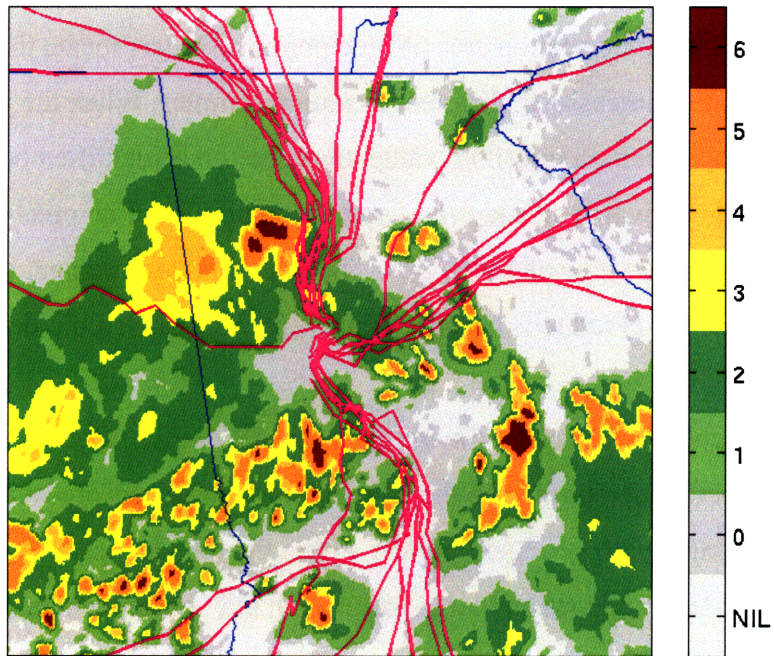


Figure 2.2. Plot of VIL and departing flights on July 22, 2006 at 21:30. A severe thunderstorm is occurring and flights maneuver away from their normal flows in order to avoid severe weather. The standard six-level scale is used to show the VIL intensity.

2.4.2 Echo Tops

Echo tops are the radar indicated top area of precipitation. Radars locate the echo top by measuring precipitation intensity at different elevation samples. When the precipitation intensity drops below a certain threshold value, the echo top is identified because there is strong precipitation below that altitude and weak precipitation (or no precipitation) above it. CIWS uses a threshold value of 18dBZ [5]. The echo top measurement tends to be slightly below the actual cloud tops because clouds are more difficult to detect by radar.

Echo tops are particularly valuable for aviation because it carries information about the nature of individual thunderstorms and the potential for causing severe weather. Higher echo tops are correlated with stronger updrafts within the thunderstorm. A strong updraft can create convective wind gusts and large hail, which may make the airspace untraversable for airplanes.

Echo tops alone cannot be used to predict the weather that pilots will choose to avoid, particularly in terminal area. While high echo tops are correlated with severe weather, it can only be accurately determined when combined with the other weather products. Echo tops and VIL are often used in combination to predict severe weather conditions.

Echo Top Plot of Thunderstorm

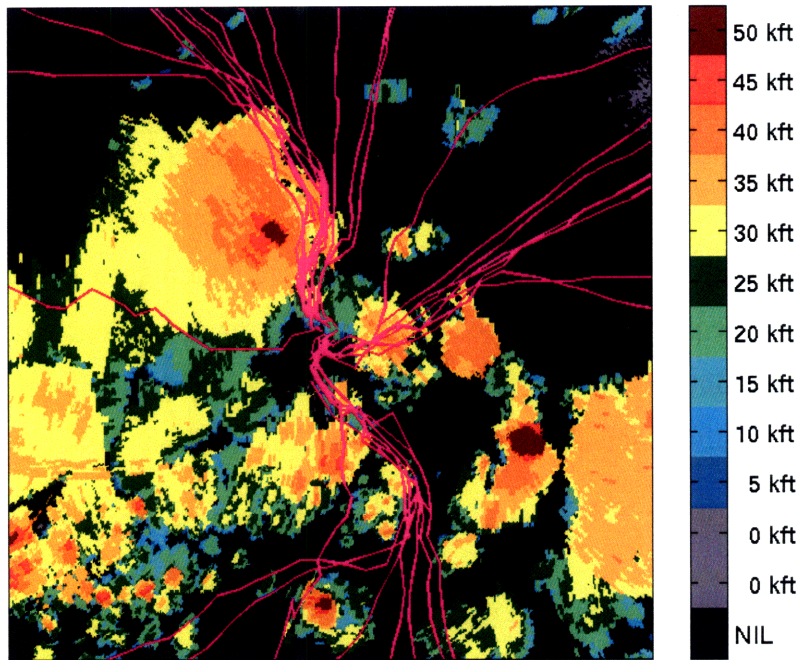


Figure 2.3. Plot of echo tops and departing flights on July 22, 2006 at 21:30. A severe thunderstorm is occurring and flights maneuver away from their normal flows in order to avoid severe weather.

2.4.3 Widest Traversable Gap

One way to estimate how likely a plane would remain inside a traffic flow is by finding the widest traversable path through a region of severe weather that avoids weather obstructions in the flow. The width is the narrowest section of the path. There may be more than one traversable path, but the widest traversable gap is assumed to be the path that pilots are most likely to take. Larger gaps are safer for air planes to travel through because it allows the pilot to have more maneuverability and gives a greater margin of safety. Also, depending on noise and the accuracy of the weather prediction, a smaller gap might not actually be safe enough to go through, and the pilot would be forced to deviate.

Finding the widest traversable gap is a technique that has been used in en route airspace to predict whether planes will deviate. Obstructions are usually defined as weather conditions that exceed a certain VIL and/or echo top level. If severe weather is present in a flow, it will always place more limitations in where the pilot can go and may make the widest traversable path narrower. Finding the widest traversable gap is more difficult in terminal airspace because traffic flows are irregularly shaped.

3. News

This chapter outlines and discusses the new techniques, methods, and original ideas that were created by this thesis. The Terminal Traffic Flow Identifier is broken down into several smaller algorithms: converting ETMS data into a more usable form, clustering the data into traffic flows, and statistically calculating the traffic flow boundaries. These algorithms and techniques are described in depth in following sections.

Also in this chapter, I introduce a new algorithm to estimate the widest traversable gap in traffic flows. Such algorithms exist already for en route traffic. This new algorithm is based on existing algorithms with some modifications in order to make it work with traffic flows.

3.1 Modeling the Terminal Airspace

Clustering the airplane trajectories into flows is a difficult problem because the trajectory data is recorded based on flight time. One data point is recorded by ETMS every minute. Trajectory data will have different lengths because planes fly at different speeds and for different distances. For example, if one plane takes 15 minutes to exit terminal airspace and another plane takes 18 minutes, then the first plane has 15 data points and the second plane has 18 data points. Each data point includes latitude, longitude, altitude, and time. In order to apply a clustering algorithm to identify flows, all trajectories need to have the same features. This is a problem because the trajectories have different numbers of data points. Also, having trajectories sampled every minute is not very helpful because two planes traveling in the same flow with identical trajectories, but traveling at different speeds, would have very different sample points. The ETMS data needs to be processed and re-sampled into something that is easier to work with.

One simplification would be to completely ignore altitude. As mentioned earlier Section 2.2, pilots are more restricted in what they can do with altitude in terminal airspace because they are either landing or taking off. A flow would not be on top of another flow because it is too dangerous to have two planes flying directly on top of each other when they are both changing altitudes. Therefore, it is possible to identify flows simply by looking at a 2-dimensional coordinate plane, ignoring altitude.

The time dimension can also be simplified. A plane generally leaves the terminal airspace in less than 20 minutes. A thunder storm is not expected to move very far in this short period of time, therefore it is not necessary to accurately keep track of where the flight is all times. Therefore it is assumed that once a plane takes off, the weather conditions are frozen and the plane travels its trajectory and leaves the terminal airspace. To do this, only the starting time is needed. This simplification prevents strange unrealistic cases from happening, such as a plane slowing down or stopping in mid air and waiting for a storm to pass before continuing forward. This assumption generally holds for large, more organized storms. However, it can be problematic for small scattered thunderstorms, where 20 minutes may be long enough for a storm cell to develop or for one to die away. This thesis will not deal with this type of weather.

Another simplification is to assume that all departing flights are moving away from the airport until they leave terminal airspace. This assumption makes logical sense because a plane that is leaving an airport should not turn around and move back towards the original airport because that would be inefficient. This assumption is generally true in Atlanta and most other large airports in the United States. However, this assumption does not hold in extremely complicated airspace such as New York, where there are multiple airports nearby and planes must take complex routes to avoid neighboring traffic. Also, it is reasonable to assume that planes will eventually leave terminal airspace. It does not make sense for a plane to take off from an airport and land back onto the same airport. This assumption implies that all traffic flows extend to the end of terminal airspace.

Finally, all trajectories are continuous and all flights will leave terminal airspace. Obviously, a plane cannot disappear and reappear in a new location. The ETMS data points are linearly interpolated to generate a continuous trajectory.

Briefly summarizing all the simplifications and assumptions:

- Altitude is ignored
- Starting time is the time of the flight
- Planes always move away from airport
- Planes always exit terminal airspace
- All trajectories are continuous

With all of these simplifications, the trajectories can be modeled using a vector of circular coordinates. With the airport as the origin of a circular coordinate system, n different circles of different radiuses r_1, \dots, r_n , with $r_1 < r_2 < \dots < r_n$ and r_n small enough to be in terminal airspace, can be drawn around the airport. All the circles are centered at the origin or airport location. Based on the assumptions and simplifications a trajectory must intersect each circle exactly once at (r_i, θ_i) . The intersections of these n circles are new re-sampled data points for the trajectories. The advantage of using these points is that all trajectories are now vectors of intersections and all the vectors have the same size n . Also, the sampling interval is no longer time, but when the trajectory reaches a certain radius, which is much more meaningful. Similar trajectories of flights at different times and moving at different speeds will have similar vectors. A clustering algorithm can be applied in an attempt to identify traffic flows.

Circular Coordinate Model for Terminal Airspace

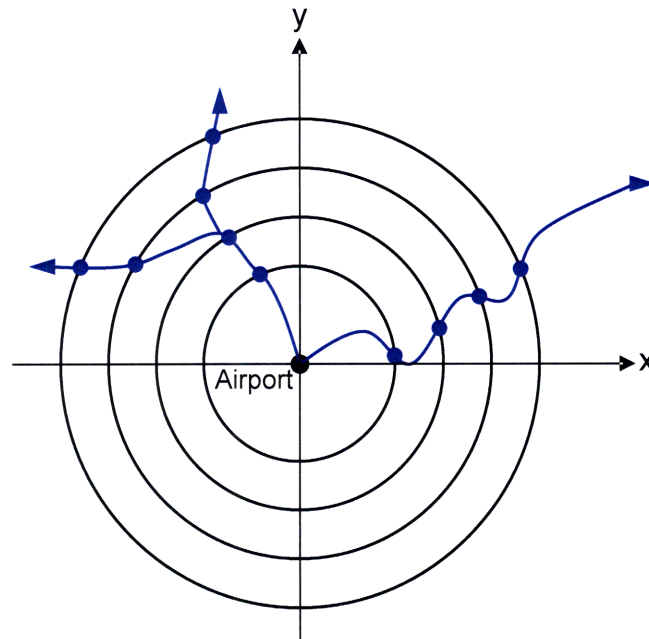


Figure 3.1. Circular coordinate model for terminal airspace. The airport becomes the origin of this new coordinate system. Many circles of different radiuses are drawn around the airport and trajectories are sampled at the point that they first intersect the circle. Because trajectories start at the airport and are always moving outward, they cross all the circles exactly once. The vector of intersections, represented as a radius and angle pairs (r, θ) , is the trajectory under this new model. Assuming enough circles are drawn, the original trajectory can be reconstructed without significant loss of information.

Technically, there can be some special cases that are not well represented in this model. These cases occur when trajectories contain large turns in between the sampling circles. These trajectories satisfies all the rules and assumptions in the model, but results in trajectory vectors that are not representative of their actual path. Fortunately, these cases do not occur in practice because of physical limitations and because pilots tend to take efficient paths.

Special Case: Trajectory Contains Sharp Turns

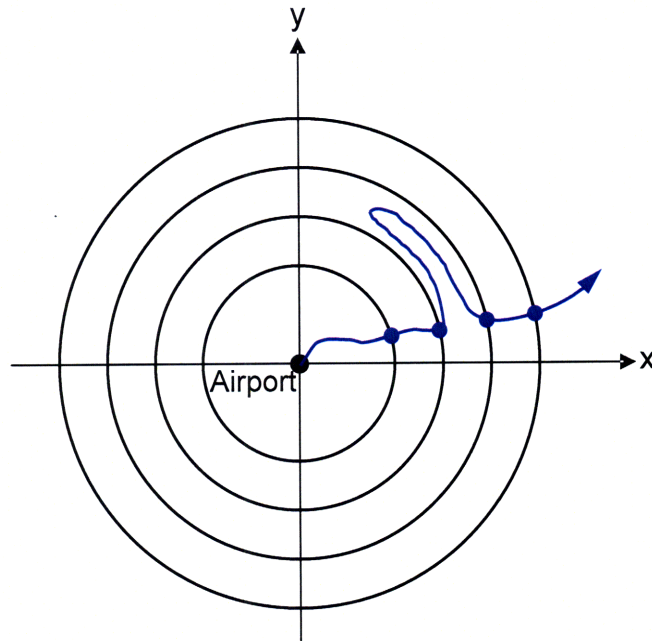


Figure 3.2. Special case where trajectory contains very sharp turns. The sample points are not representative of the trajectory the plane actually traveled. This problem can be avoided by making sure the sampling circles are fine enough. Planes have physical limitations on turning ability. For example, a plane physically cannot make a 90 degree while traveling at hundreds of miles per hour. Assuming that the distance between the radiuses of the sampling circles are sufficiently small, this situation should never occur.

Special Case: Spiral-like Trajectories

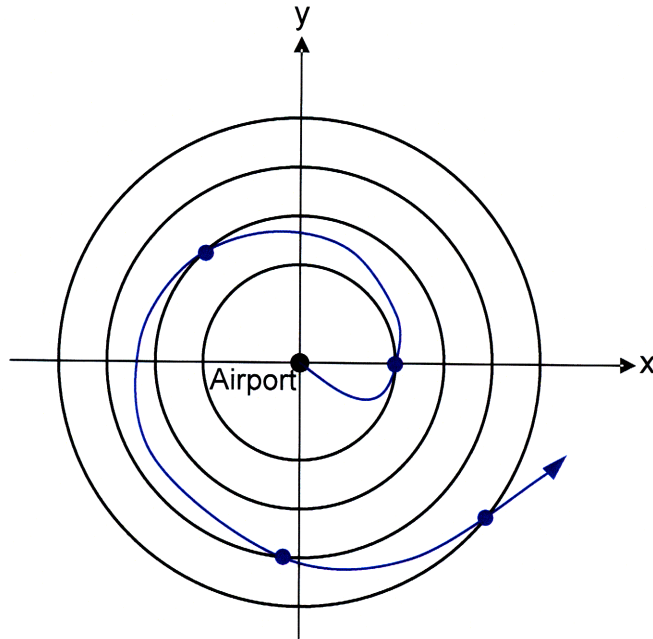


Figure 3.3. Special case where trajectory contains spiral-like movements. This type of trajectory is physically possible for a plane to fly. In this special case, the reconstructed trajectory from the sample points on the circles would look very different from the actual one. However, logically, this type of trajectory is inefficient because it is very long and it uses up a lot of airspace. Trajectories like these tend not to occur.

The data originally comes in (latitude, longitude) pairs. This can be converted into (x,y) coordinates through a map projection, which projects the spherical surface of the earth onto a flat two dimensional coordinate system. The CIWS map projection is used because it will project the trajectories onto the same coordinate system as the weather data, which helps avoid confusion and ensure that trajectories will be correctly mapped onto CIWS weather fields.

3.2 Clustering Algorithm to Identify Flows

After converting trajectories to the new circular coordinate model, all trajectories are essentially vectors of the same length. A set of these trajectories can be represented as a matrix, with rows representation flights and columns representing the different discrete

circles in the model of the airspace. The values in each row are the intersection angles in the trajectory vector.

Set of Trajectories Represented as a Matrix

r_1	r_2	r_3	r_4	r_5	
0.122	0.123	0.124	0.122	0.221	
1.444	1.443	1.450	1.455	1.501	...
0.987	0.982	0.980	0.984	0.990	
		⋮			
θ_1	θ_2	θ_3	θ_4	θ_5	

Figure 3.4. A set of trajectories can be represented as a matrix. Each row represents one flight. Each column represents the circles of different radiuses r_1, \dots, r_n . The values inside the matrix are the angles, shown in radians in the example above, that the trajectory intersects the circle. Therefore, all the (r, θ) pairs that make up trajectories are stored.

Trajectories that belong to the same traffic flow should be very similar, meaning that they should intersect the circles at about the same angles. If a group of trajectories belong to the same flow, then in the matrix representation, the columns should contain angles that are very close to each other. Clustering algorithms automatically go through this matrix and partition the rows into clusters such that differences in the numbers in each column is very small. This would identify similar trajectories that are likely to be members of the same traffic flow. The columns are known as features or attributes because they contain identifiers that determine whether items are similar or different.

Matrix of Trajectories Being Clustered

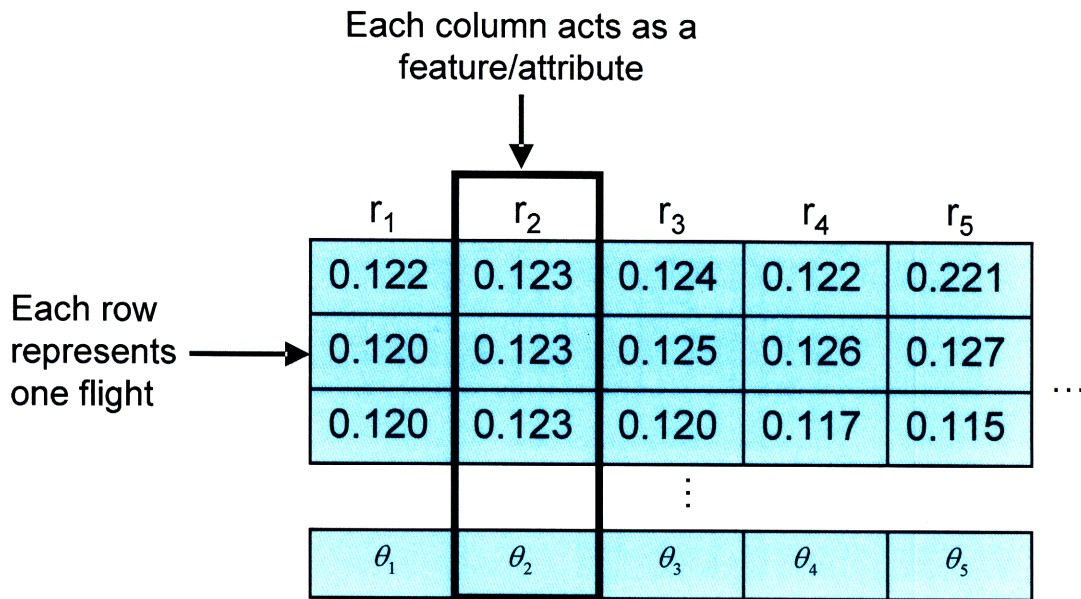


Figure 3.5. Clustering algorithms uses the columns as features or attributes that determine if the trajectories are similar. Clustering algorithms attempt sets of trajectories such that the columns in this new set would be very similar in value.

It would be nice if one of the popular clustering algorithms, such as k-means, would be able to identify traffic flows. Unfortunately, none of the popular clustering algorithms work very well on this problem. The number of air traffic flows that is being used varies all the time. This produces a problem for many clustering algorithms. Also, because a lot of flight data is available, it is desirable to have an algorithm that runs very quickly and efficiently. Flow identification probably requires tens of thousands of trajectories to be sufficiently accurate.

This section reviews two popular clustering algorithms, the k-means and hierarchical algorithms, and discusses the strengths and weaknesses of each with regards to identifying traffic flows. Then a new hybrid algorithm that combines some of the properties of both k-means and hierarchical clustering is introduced. This hybrid algorithm is much better suited for identifying traffic flows than the others.

3.2.1 K-Means

K-means is an algorithm that clusters n objects into k partitions based on attributes. The algorithm estimates the centers of natural clusters, or centroids, in the data. Usually, k-means attempts to minimize the total intra-cluster variance, or total squared error,

$$E = \sum_{i=1}^k \sum_{x_j \in s_i} (x_j - \mu_i)^2$$

where s_i are clusters with $i = 1, 2, \dots, k$ and μ_i is the centroid, or the mean of all points $x_j \in s_i$. However, different error functions may be used.

K-means belongs to a family of algorithms known as partitional clustering algorithms. Partitional clustering algorithms attempt to determine all the clusters at once and refine the results with iterations.

The k-means algorithm is usually implemented in the following way:

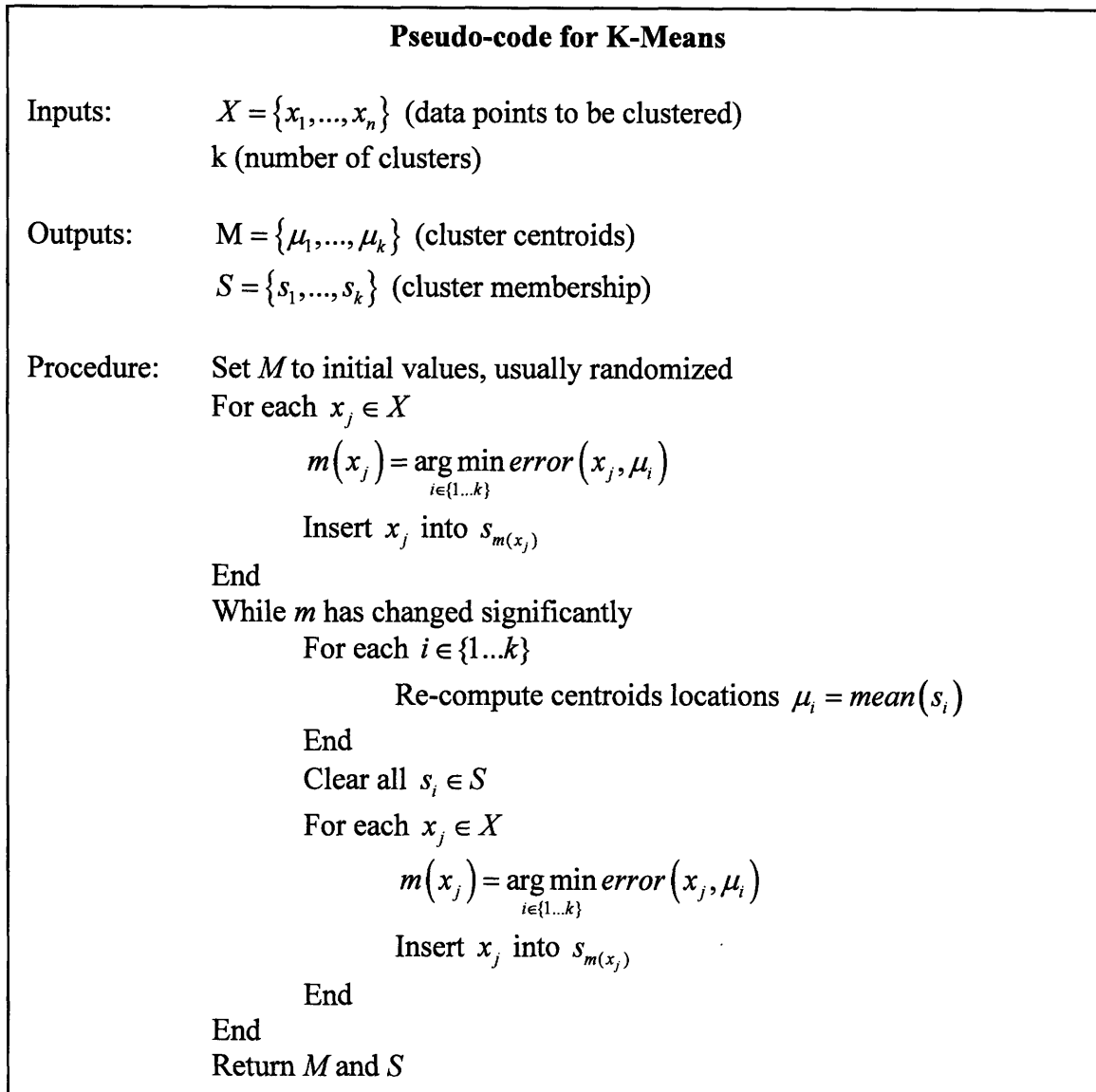


Figure 3.6. Pseudo-code for k-means clustering algorithm.

One weakness of the k-means algorithm is that it is not guaranteed to give optimal results. Non-optimal results may occur when the data points are arranged in such a way that there are multiple local minima for the total error function. Sometimes, a small cluster of isolated points occur in a dataset. If a centroid starts near this cluster, it may converge onto the cluster, even though that is not optimal global minimum error.

Because k-means does not always find the optimal solution, k-means may give inconsistent results. The starting points are randomized and that determines where the

centroids will eventually converge onto. If consistent results are important, then k-means might not be a good choice.

K-means has a best case running time of linear and a worst case running time of super-polynomial [6]. Fortunately, in practice, the running time tends to be towards the lower end of these boundaries. K-means is one of the most popular clustering algorithms because it tends to run fast on real experimental datasets. The problem of not always finding the optimal solution can sometimes be avoided by running k-means many times on the same dataset, and then choosing the best solution. This will increase the probability of finding the optimal solution or a solution that is good enough. Running k-means multiple times is still usually faster than most other clustering algorithms.

The biggest weakness of k-means is that the number of clusters k needs to be known beforehand. If the value selected for k is too small, clusters would be too big and have large variances. In terms of traffic flows, this would result in a few flows that are very wide. If k is too large, then there will be too many centroids and they would be too close together. In terms of traffic flows, this would result in many small flows that are touching each other. K-means can give poor results if the number of clusters in the dataset is unknown. Unfortunately, the number of traffic flows in a set of trajectories is unknown, so it would be difficult to use k-means to identify these flows.

One possible solution to find the number of clusters is to run k-means on a range of values for k and then heuristically select one that works well. This will increase the run time by a constant multiple. When working with large datasets, such as tens of thousands of trajectories, one run of k-means takes substantial time (on the order of many seconds to minutes). Attempting a large range of k values may have significant effects on usability. While this is a feasible solution, the hybrid algorithm described later in Section 3.2.3 runs faster.

3.2.2 Hierarchical Clustering

Hierarchical clustering is a family of clustering algorithms that forms clusters by combining smaller clusters. The results iterations are stored in a tree structure called a dendrogram. There are several different types of hierarchical clustering algorithms. The most basic and common one is agglomerative hierarchical clustering. Here is how the algorithm works:

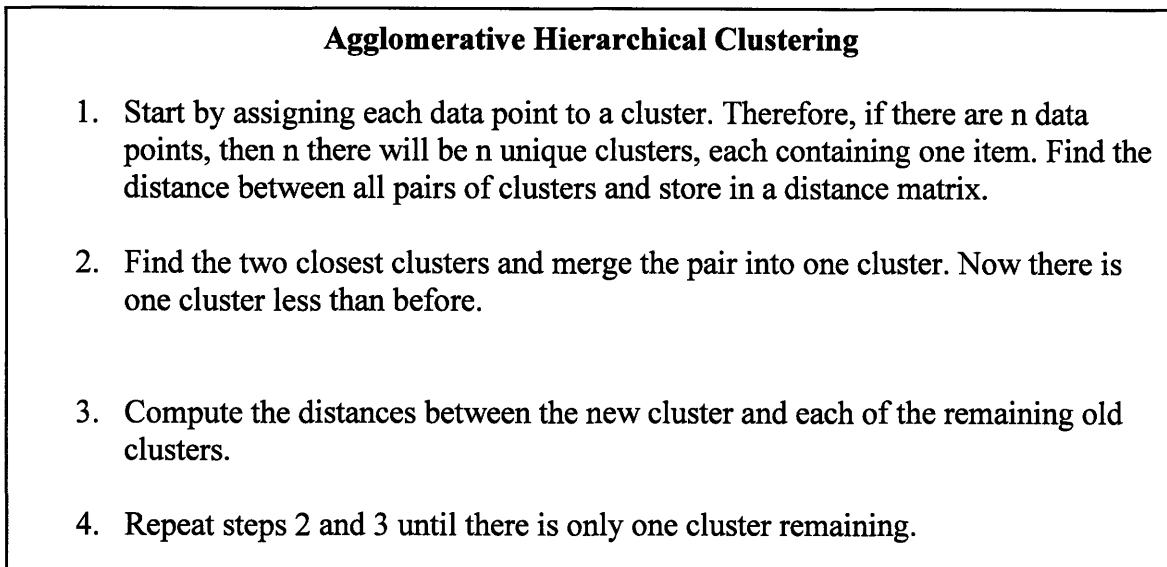


Figure 3.7. Outline for the agglomerative hierarchical clustering algorithm.

Note that the goal is not to get one big cluster, which is the stopping condition. That can be done trivially. When the algorithm ends, the entire dendrogram will be created. From the dendrogram, it is possible to extract the clustering for a specific number of centroids simply by looking at a specific level of the tree.

Heuristics can be applied to the dendrogram to try to determine the number of natural clusters if it is not known. Usually this involves examining the total intra-cluster variance and seeing how that changes if clusters are added or removed. If there are k natural clusters, the variance is not expected to decrease much when changing from k clusters to $k+1$ clusters. However, the variance will be much larger when changing from k clusters to $k-1$ clusters. Another heuristic that can be applied is simply a distance threshold, where

all clusters are assumed to be at least a certain distance apart. The largest k that satisfies this distance constraint would be the solution.

The main weakness of agglomerative hierarchical clustering is that it has a quadratic runtime and requires quadratic space for the distance matrix. Other variants of hierarchical clustering also have very long runtimes. Therefore, for very large datasets, it might be unfeasible to do hierarchical clustering. Unfortunately, identifying terminal traffic flows uses extremely large datasets. Trying to find all the flows used in Atlanta for a week would require clustering tens of thousands of trajectories. With quadratic runtime and quadratic space, it becomes too difficult to handle larger datasets.

3.2.3 Hybrid K-Means/Hierarchical Clustering

K-means and agglomerative hierarchical clustering both have weaknesses that make it not well suited for identifying air traffic flows. K-means requires the number of clusters to be known beforehand and agglomerative hierarchical clustering is too slow and requires too much space. However, k-means has the advantage that it runs fast and hierarchical clustering has the advantage that it is easier to heuristically determine the number of clusters in a dataset. A clustering algorithm that has the strengths of both and weaknesses of neither is desired.

A hybrid k-means/hierarchical clustering algorithm would fit the criterion. The hybrid algorithm starts by performing k-means with a large k . This creates too many clusters. Then the clusters are merged with agglomerative hierarchical clustering steps.

Hybrid K-Means/Hierarchical Clustering

1. Perform k-means with a large k and random initial centroids.
2. Find the distance between all pairs of resulting centroids.
3. Merge the two closest centroids. Place this centroid at the mean of the members of the clusters being merged. Now there is one less centroid.
4. Perform k-means on the remaining centroids to make sure they are centered. Do not randomize starting locations. Keep the centroid locations from the previous step.
5. If the resulting centroids meet the stop criterion, then and return the centroids and cluster memberships. Otherwise repeat steps 3-5.

Stop criterion: All pairs of centroids must have a distance between them that exceeds a specified threshold.

Figure 3.8. Outline for the agglomerative hierarchical clustering algorithm.

Note that we are clustering angle measurements on a circular coordinate system, so the standard distance or squared distance error function would not work for k-means. The error function should be the angle apart,

$$E = |\theta_i - \theta_j|$$

where E is an angle with $E \in [0, 2\pi]$. Alternatively, E^2 may be used if square error is desired, but both tend to yield the same results in practice.

A good value for the large k would be about two times the number of clusters that would be expected. For example, there are about 25-30 traffic flows for departing flights in Atlanta, so a k value of 50-60 would give good results. A k value that is too large would result in excessive work. However, it should be large enough such that there is confidence that it exceeds the number of natural clusters in the dataset.

The hybrid algorithm is slower than k-means, but not by too much. On the first step, it performs the entire k-means algorithm. On successive performances of k-means (Step 4

in Figure 3.2.3.1), the running time is actually very fast because old centroid clusters are used. Most of the time, the centroids are already converged and nothing needs to be done, making the hybrid algorithm much faster than attempting k-means over a range of k values. This re-clustering step mainly serves as a check for special cases. In practice, it can probably be omitted without any changes to the outcome. However it runs quickly enough that it can be left in without any significant affect to running time. The hierarchical steps of the hybrid algorithm that merge centroids have quadratic running time and space with respect to the number of clusters. Assuming that there are a very small number of clusters compared to the amount of data, these steps are insignificant. Also, k-means generally requires multiple runs to make sure that it did not get stuck at particularly bad local minimum. The hybrid algorithm does not need multiple runs because it starts off capturing too many clusters, so it probably captures all or most of the mini-clusters that cause local minimums. These clusters are then merged in a predictable fashion, so the hybrid algorithm tends to give fairly consistent results, even though consistency is not always guaranteed.

There are many ways to decide on a stop criterion, such as by how the variance changes between iterations. However, a distance measure seems to work well for identifying traffic flows. The centroids of traffic flows are expected to have some significant distance between them because it is too dangerous for planes to fly too close together. After experimenting with various error functions and visually examining solutions, the best stopping criterion is simply the square error at all sample points. If centroid A is a trajectory vector containing values a_1, \dots, a_n and centroid B is a trajectory vector containing values b_1, \dots, b_n , then the error between A and B is,

$$Error(A, B) = \sum_{i=1}^n (a_i - b_i)^2$$

And the stop criterion would be satisfied when the square error exceeds a threshold value for all pairs of centroids. The threshold can be determined experimentally with trial and error. After a good threshold is obtained, it can be used over and over again assuming the same set of sampling radiuses (discussed in Section 3.1) are used for trajectories.

3.3 Estimating Flow Boundaries

The hybrid clustering algorithm partitions the dataset into clusters and returns the centroids of the clusters. The centroid, which is the mean of all the trajectories in the cluster, is the best guess at the path that pilots were trying to follow. Looking at how far the trajectories within the cluster moves away from the centroid should give an idea of how strictly the pilots must follow the trajectory.

It is not a good idea to simply take all the trajectories and draw boundaries so that all the trajectories are enclosed and call that the boundaries of the flow. There may be a few samples that were clustered incorrectly or there may be some flights that are simply very irregular that would throw off the boundaries. These trajectories should be treated as noise data. To determine the boundaries of the flow, the noisy trajectories need to be removed and then boundaries can be drawn such that it contains all the remaining trajectories.

Visually examining the flows, it seems like there is some Gaussian-like properties to them. There is a heavy concentration of trajectories that follow the centroid almost exactly. There are fewer trajectories that occur further away from the centroid. This pattern is a lot more obvious when looking at the intersections of a cluster of trajectories and one of the circles around the airport.

Flights Crossing Circle Around Airport

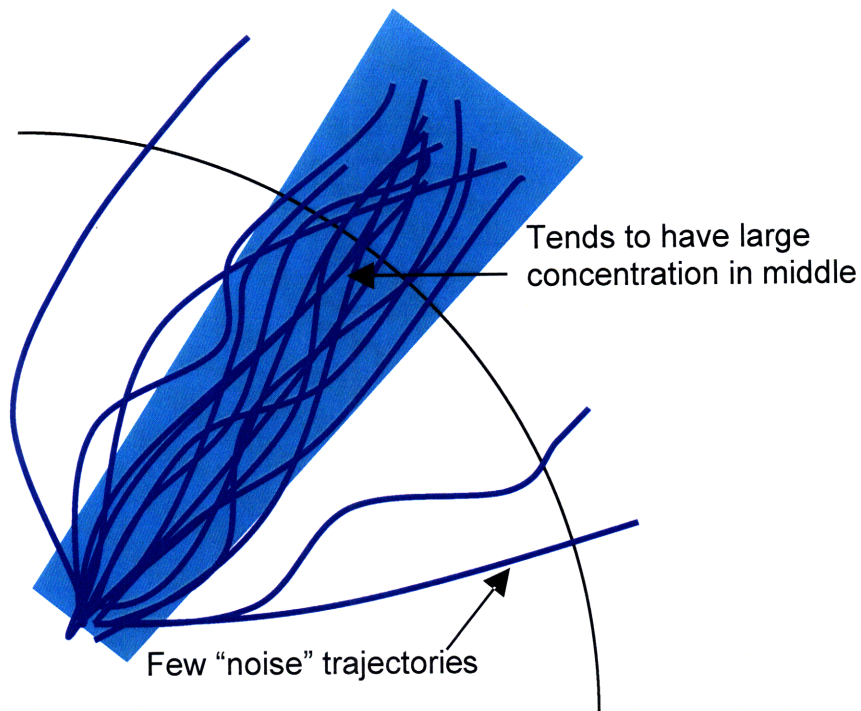


Figure 3.9. Stylized diagram of how trajectories tend to cross one of the circles around the airport, or origin.

Simply examining the cluster trajectory and centroid trajectory at the points that it crosses the circles around the airport will make this more obvious. Looking at the (r, θ) coordinates of the trajectories and centroid at a given r , a histogram can be made of $\theta_{traj} - \theta_{centroid}$, or the angle difference between the cluster trajectories and the centroid.

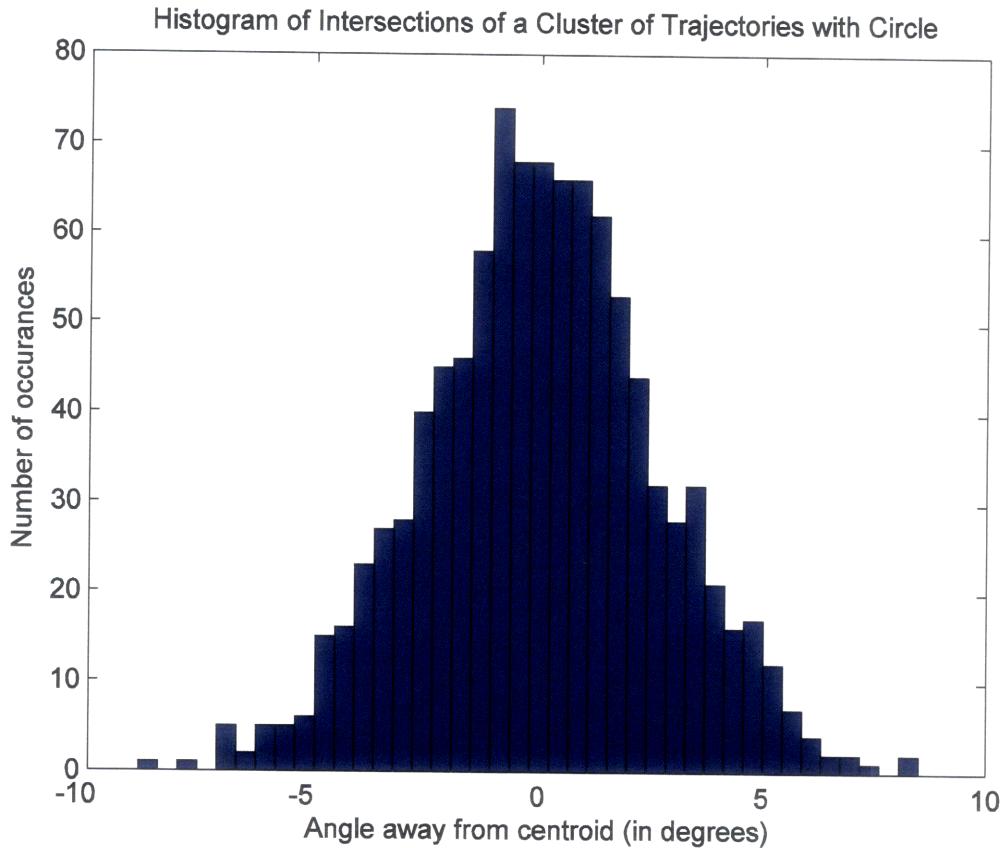


Figure 3.10. Sample histogram of the angle difference between the centroid and cluster trajectories. Most trajectories are very close to the centroid. The distribution is Gaussian-like.

As seen in Figure 3.3.2, most trajectories in a cluster tend to only be a few degrees away from the centroid. Also, the shape of the histogram is Gaussian-like. A reasonable way to determine the boundaries of the flow would be to examine the histogram at every radius level and remove a few percent from both extremes. Repeating this for every radius level would create several arcs that trajectories must go through in order to stay inside the trajectory. This forms the boundaries of a flow.

3.4 Widest Traversable Gap Algorithm

Finding the widest traversable gap is a difficult problem because of all the different directions a plane can fly. Without creating some simplifications, this problem is very difficult to solve quickly. The algorithm described in this section is a modified version of an existing algorithm used to find widest traversable gaps in en route airspace [7].

Since the CIWS weather data has one kilometer resolution, a flow can be divided into one kilometer blocks. A rectangle can be drawn around the flow and rotated so that the flow looks like the matrix in the following figure:

Flow Projected onto Matrix with Weather

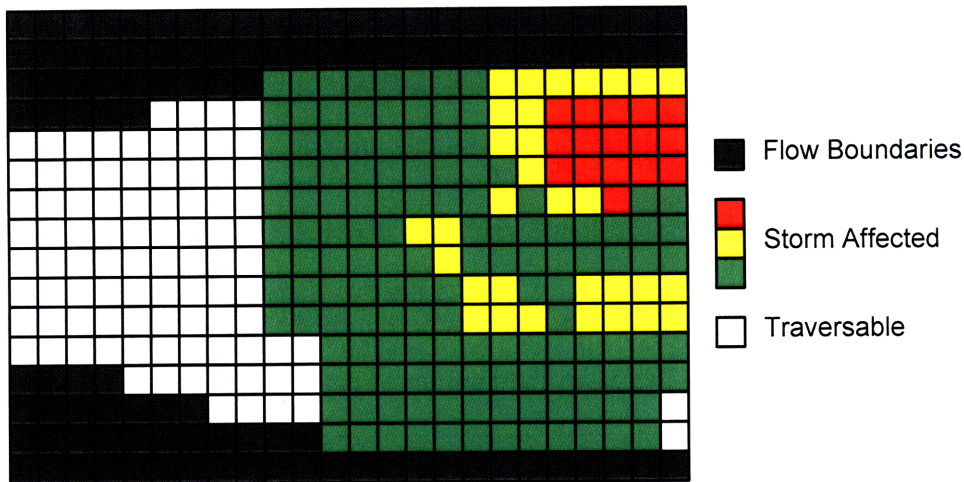


Figure 3.11. A traffic flow has been rotated and projected onto a matrix. Weather data is also added. For simplicity, in this figure, the weather can only be low, medium, or high intensity (green, yellow, and red respectively). Planes travel from the left side to the right side.

On this matrix, the airport is on the left side and the flights are trying to move to the right side without exiting the flow boundaries. If a flight cannot reach the right side due to weather conditions, then the plane must leave the flow in order to reach its destination. In the example shown in Figure 3.4.1, assume that the yellow and red storm regions are blocked and impassible. Also, assume that planes cannot move backtrack or move left,

which is reasonable because planes physically cannot fly backwards and all trajectories are moving away from the airport. They can only move to the right, up, or down on the matrix.

Matrix of Traversable Regions

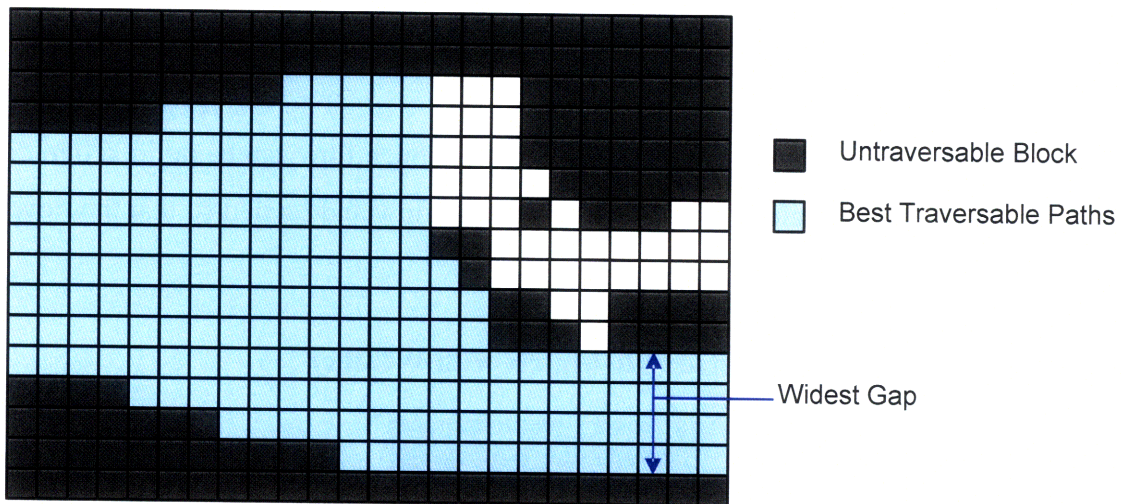


Figure 3.12. Matrix of traversable and in-traversable regions. Assuming plane starts on the left side, and can only move to the right, up, or down on traversable blocks, the paths that can potentially lead to the widest traversable gap are highlighted.

Even with all these assumptions, there are still a large number of paths to check. Each additional column on the matrix increases the number of paths at a super-polynomial rate. Shortest path algorithms do not work because there is no distance measure. On each move through the matrix, the gap width can be calculated by examining the number of traversable blocks in the horizontal or vertical direction that are connected. The cost of going through a path is the minimum gap width encountered on the path.

A further simplification is to not make all the right, upward, and downward connections between traversable blocks. The following figure shows how to make connections that will likely preserve a path that has the widest traversable gap.

Reduced Matrix of Traversable Regions

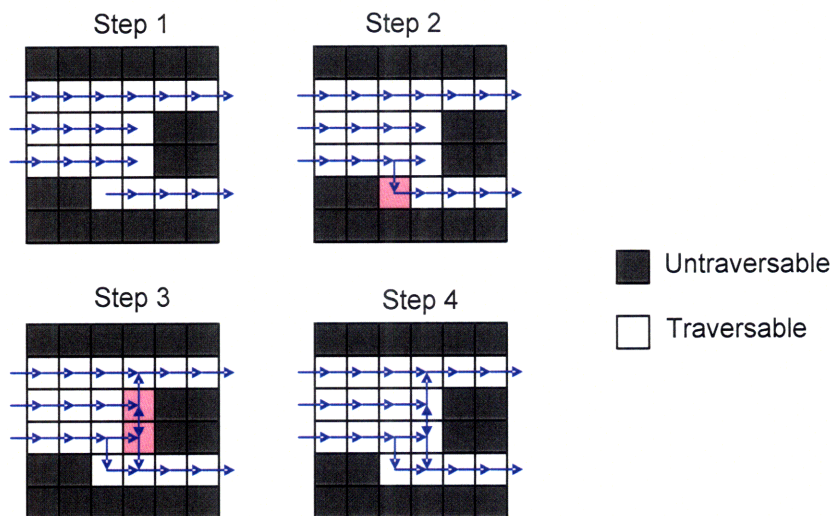


Figure 3.13. Matrix of traversable regions with fewer connections. Connecting the blocks in this manner reduces the total number of paths while likely preserving a path that contains the widest traversable gap. Step 1: Make connections pointing to traversable blocks on the right. Step 2: For traversable blocks that cannot be landed on (highlighted above), make connections to it from adjacent traversable block from the top or bottom if available. Step 3: For traversable blocks that have in-traversable blocks directly on the right (highlighted above), make connections upward and downward. Step 4: All connections are complete. This is the new matrix of traversable regions.

The new matrix with reduced connections contains many fewer possible paths. It is usually fast enough to just examine all these trajectories using an exhaustive search to find the widest traversable gap. There are some special cases where the layout of in-traversable blocks causes many trajectories to be generated and an exhaustive search would take too long to complete. Also, there may be some special cases where the layout causes the actual widest traversable gap to be lost. However, actual weather data tends to have simple contour-like shapes rather than maze like patterns. Strange patterns of in-traversable blocks that cause problems rarely occur.

3.5 Application and Analysis

Collectively, the techniques used to resample the ETMS data, identify air traffic flows, and find flow boundaries make up the Terminal Traffic Flow Identifier algorithm. This algorithm can go through a database of ETMS trajectory data to characterize terminal air traffic flows.

Typically, to perform a weather analysis, air traffic flows and boundaries for clear days need to be identified. It is important to know the variations of trajectories in clear weather to determine if deviations are actually caused by weather. Accurate traffic flows and boundaries can be generated by applying the Terminal Traffic Flow Identifier algorithm on a very large number of flight trajectories. To demonstrate the algorithm, 20,000 departing flight trajectories from Atlanta from seven different clear weather days were used to generate traffic flows and boundaries. This statistical approach of analyzing large amounts of flight trajectories created more complex shaped flows that are more representative of the airspace pilots actually use than the models that currently exist.

Trajectories and Flow Boundaries

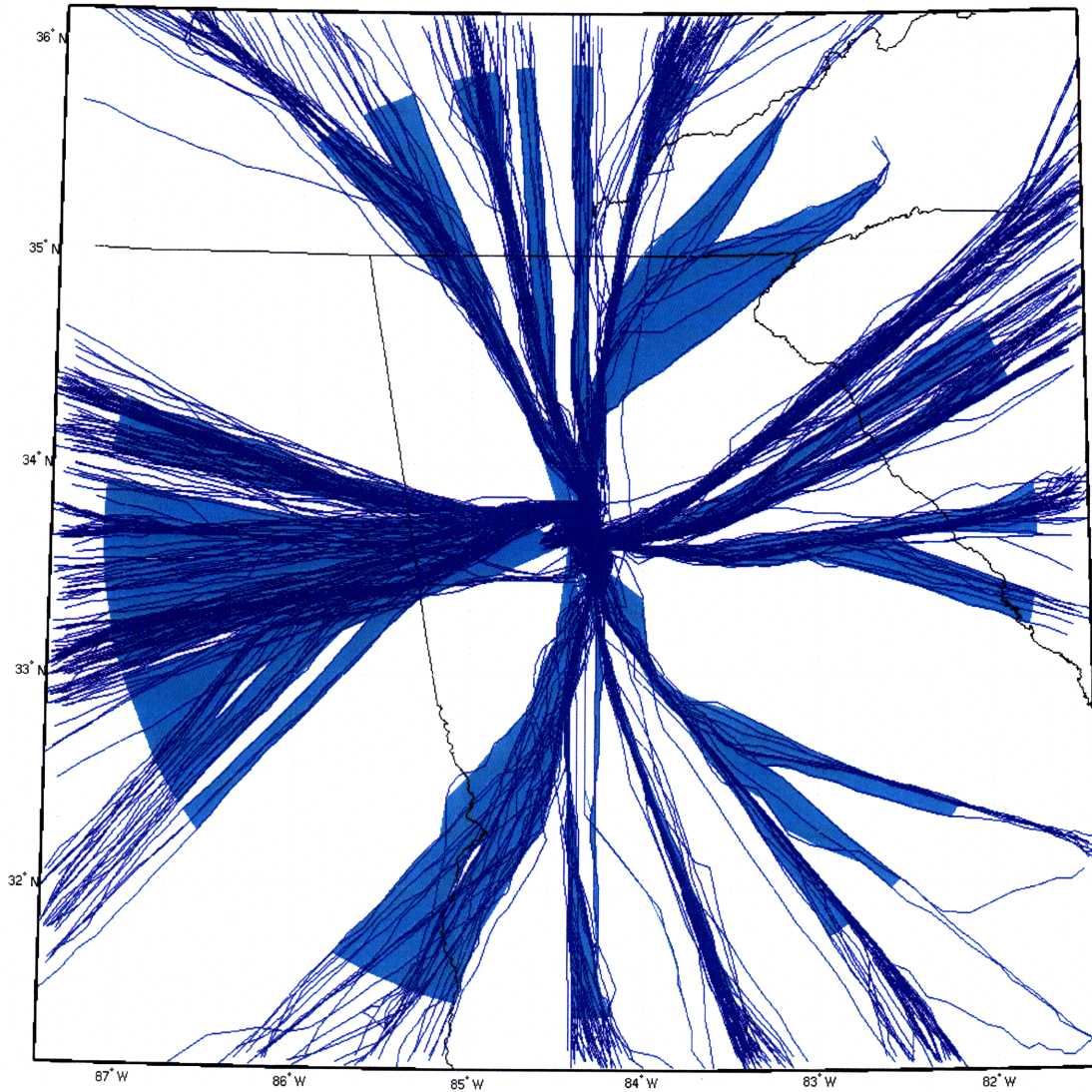


Figure 3.14. Plot of trajectories and traffic flow boundaries for departing flights in Atlanta on clear days. All departing trajectories from seven clear days, consisting over 20,000 flights, were used to estimate the flows and boundaries.

As a demonstration of how the algorithms developed in this thesis can be used, sample cases for when trajectories stayed on course during a storm and when trajectories deviated to avoid a storm were gathered from the ETMS database. Several measurements can be taken from the flow at the time the flights took off: the average VIL and echo top, the minimum VIL and echo top, the maximum VIL and echo top, the percentage of the flow area that is affected by the storm, and the widest traversable gap at various VIL and

echo top thresholds. These measurements combined with whether the flight deviated or stayed on course can be used to train a neural network. MATLAB provides a neural network toolbox that automatically implements various types of neural networks.

Flows, VIL, and Echo Tops during a Thunderstorm

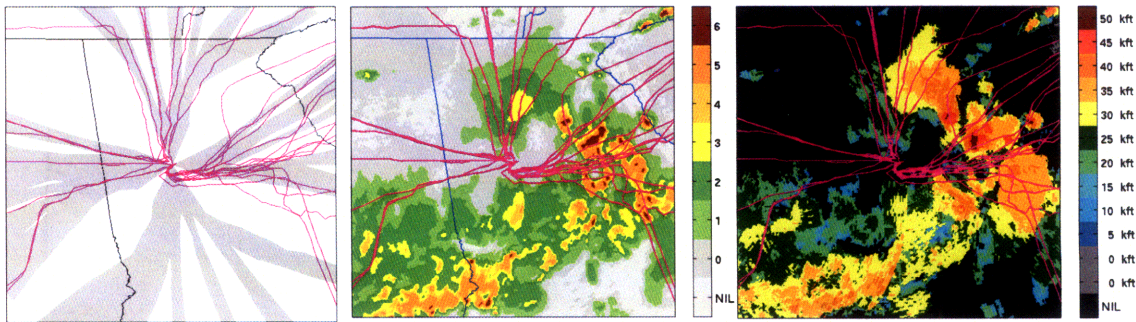


Figure 3.15. Plots of normal traffic flows, VIL, and echo tops, all with trajectories on top. The first plot shows the shape of normal clear day trajectories. Some flights are able to penetrate the storm and follow the flow while others need to exit the flow. Characteristics about the weather within the flow boundaries may determine why some flights were able to penetrate the storm and why others cannot. These characteristics include the distribution of the weather (mean, maximum, minimum, etc. of VIL and echo tops) and the widest traversable gap.

A neural network examines a set of training data, which contains deviations and non-deviations, and finds the differences in the characteristics between the two. After a neural network is trained, it can give a best estimate of whether a given input set of characteristics would result in a deviation or non-deviation trajectory. This trained neural network is useful, because when given a weather forecast of the future, it can make a good statistical guess at which traffic flows would be shut down and which would remain operational.

Given the number of characteristics that can be used in the neural network many training cases are needed for statistical significance. Given time constraints on this thesis, I was only able to manually select and parse a small set of trajectories, about 150 samples. The inputs to the neural network that I used are: the maximum VIL and echo top values inside the flow, the average VIL and echo top values inside the flow, the percent of the

flow area that is under intense weather (VIL level 3 or higher), and the widest traversable gap for various VIL and echo top values.

While the results are not statistically significant, with only 150 samples, there are some promising discoveries that are worth mentioning. The VIL measurements and the widest traversable gap at VIL levels three or higher seemed to be very good predictors of whether flights will need to deviate or not. Echo tops seem less correlated. My best explanation is that in terminal airspace, flights are at a lower elevation, so they are always affected by the storm even if the storm is fairly low. Therefore, VIL becomes the more dominant predictor. These findings agree with another study done on the New York airspace using the Route Availability Planning Tool (RAPT) algorithm [8].

In terms of predicting whether individual flights would deviate from the flow or not, the neural network is correct about 80% of the time (note that this is done on a small data set, so it may not be a very good indicator of how well or poorly it can actually perform). Looking at Figure 3.5.1, this is somewhat expected because even on flows that are impacted by weather, a few flights still penetrate while the rest deviate. Through observation it seems like most of the time when the neural network guesses a deviation, at least some of the flights in flow will deviate. So the classifier does seem like it can do very well in predicting whether a flow will be affected (some flights will deviate), but only about 80% on predicting whether a specific individual flight will deviate.

4. Contributions

This thesis makes several new contributions to air traffic manage. A novel approach was used to model the terminal airspace. Representing trajectories as intersections of circles around the airport makes new forms of analysis, such as clustering, possible. Most other models do not use the fact that real flight paths are limited in how they can move, so they contain additional complexities. The circular coordinate model will be helpful in future works that analyze terminal airspace.

The Terminal Traffic Flow Identification Algorithm was developed to automatically determine flows and flow boundaries. The algorithm was used to analyze over 20,000 flight trajectories in Atlanta to accurately identify terminal area air traffic flows. The boundaries determined using this algorithm is much more representative of what pilots will actually follow. These flows allow for a much more accurate weather analysis.

This thesis also creates the tools and framework for extracting weather features from trajectories and traffic flows discovered by the Terminal Traffic Flow Identifier algorithm. A demonstration of how these features can be used with a neural network was performed with 150 flight trajectories and the conclusions agree with existing research. A statistical analysis using many flight trajectories has never been performed on terminal airspace in the past and this demonstration proves that such an approach is feasible and can give good results. The code written in this thesis will be useful for future projects and research at MIT Lincoln Laboratory.

5. References

- [1] Robinson, M. J., J. Evans, B. Crowe, D. Klinge-Wilson, S. Allan. "Corridor Integrated Weather System Operational Benefits 2002-2003: Initial Estimates of Convective Weather Delay Reduction", MIT Lincoln Laboratory Project Report ATC-313, 2004.
- [2] "Glossary of Meteorology", American Meteorological Society, 2000. May 4, 2008 <<http://amsglossary.allenpress.com/glossary>>
- [3] Robinson, M. J., James Evans, Bradley A. Crowe. "En Route Weather Depiction Benefits of the NEXRAD Vertically Integrated Liquid Water Product Utilized by the Corridor Integrated Weather System", 10th Conference on Aviation, Range and Aerospace Meteorology, American Meteorological Society, Portland, OR, 2002.
- [4] Troxel, S.D., and C.D. Engholm. "Vertical reflectivity profiles: Averaged storm structures and applications to fan beam radar weather detection in the U.S.", 16th Conference on Severe Local Storms, American Meteorological Society, Kananaskis Park, Alta, Canada, 1990.
- [5] Smalley, David, Betty Bennett and Margita Pawlak. New Products for the NEXRAD ORPG to Support FAA Critical Systems, Long Beach, CA, 19th American Meteorological Society International Conference on Interactive Info Processing Systems for Meteorology, Oceanography and Hydrology, 2003.
- [6] Arthur, David and Sergei Vassilvitskii. "On the Worst-Case Complexity of the k-means Method", 2005. May 2, 2008 <<http://dbpubs.stanford.edu:8090/pub/2005-34>>
- [7] Martin, Brian. "Model Estimates of Traffic Reduction in Storm Impacted En Route Airspace", AIAA 7th Conference on Aviation Technology, Integration and Operations, Belfast, Ireland, 2007.
- [8] DeLaura, Rich, Michael Robinson, Russell Todd, Kirk MacKenzie. "Evaluation of Weather Impact Models in Departure Management Decision Support: Operational Performance of the Route Availability Planning Tool (RAPT) Prototype", 13th Conference on Aerospace, Range and Aviation Meteorology, New Orleans, LA, 2008.