AN INTERACTIVE 4D VISUALIZATION SYSTEM FOR AIR TRAFFIC CONCEPT ANALYSIS

Andrew Crowell
Andrew Fabian
Nicole Nelson
Federal Aviation Administration, Atlantic City Int'l Airport, NJ

Abstract

The Federal Aviation Administration (FAA) and supporting organizations are developing new concepts to improve efficiency and safety of the National Airspace System (NAS). Some of these concepts change current restrictions, introduce new maneuvers, or provide better decision support to air traffic controllers. Each of these concepts requires extensive research and validation before it can be integrated into the NAS.

Many different analysis tools are used to determine benefits, limitations, and requirements of these concepts. Some of these tools provide statistical metrics of the concepts, while others show the concepts visually. However, many of these are specialized tools built specifically for analyzing one concept or a small subset of concepts in a specific region of the NAS. When a new concept is introduced, it is often difficult to find a tool that can easily be adapted to help analyze it.

The Concept Analysis Branch at the FAA is tasked with analyzing many of these concepts for feasibility and benefits determination before requirements development occurs. In order to assist in this task, the Concept Analysis Branch has developed a flexible, extensible, interactive 4D visualization tool for analysis of virtually any aviation concept. Flexible Flight Traffic Exploration Visualization 4D (FliteViz4D) was created as a unique development platform specifically directed at visualizing air traffic concepts in three-dimensional space, with two additional temporal dimensions for current time and future time. It allows concepts such as 4D Trajectory Based Operations, National Convective Weather Forecast, and 3D Path Arrival Management to be visualized in all their dimensions. This was accomplished by using object-oriented design concepts in the development of FliteViz4D to provide unparalleled flexibility and extensibility. New concepts can be added into the visualization using plugins with no changes to the existing platform. The user can decide which plugins to use, determining exactly how FliteViz4D runs and what is visualized. The result is a powerful visualization and analysis system, scalable from a single airport to the entire world, which can be used to analyze any concept that could possibly be visualized abstractly or physically.

This paper will provide an overview of the system, providing several examples of how it has been used in recent studies, how it will be used in future studies, and how it can be utilized by other organizations in the aviation industry.

Introduction

The Federal Aviation Administration (FAA) and supporting organizations are developing new concepts for improving the efficiency and safety of the National Airspace System (NAS). As part of the Next Generation Air Transportation System (NextGen) many of these concepts introduce new maneuvers, modify current restrictions, and provide improved decision support tools to air traffic controllers. Before any of these concepts are integrated into the NAS, extensive research and validation is required to determine benefits, limitations, and requirements. Many different analysis tools are available to help in this process, some of which provide statistical information, others which provide visualization. However, most of the visualization tools are designed specifically to analyze a particular concept or a group of concepts. When a new concept is introduced, it is often difficult to find a tool that can be adapted to visualize this new concept.

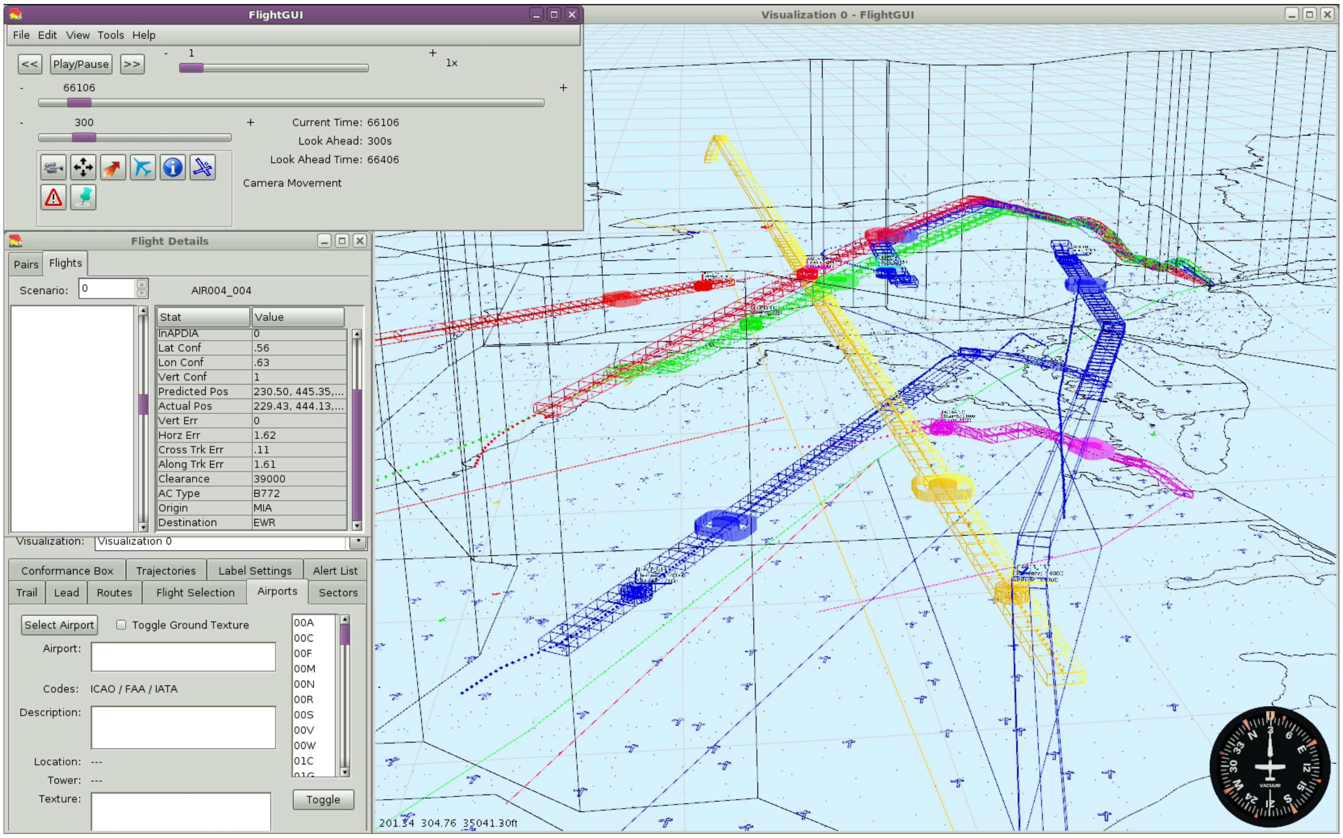

Figure 1. Typical display of FliteViz4D software

The Concept Analysis Branch (CA) at the FAA is tasked with analyzing many of these concepts before the requirements phase. In order to assist in analyzing these concepts, CA has developed a flexible, extensible, interactive 4D visualization tool for analysis of virtually any aviation concept.

The project was named Flight Graphical User Interface (FlightGUI) [1] and has been released to the public as a partially open-source software platform under the name Flexible Flight Traffic Exploration Visualization 4D (FliteViz4D) (Figure 1). The purpose of this software is to provide an unparalleled visualization and analysis platform to the aviation industry for analysis of concepts, post-analysis of air traffic, and presentation of air traffic information.

Background

The FlightGUI project was initiated in 2005 by the Concept Analysis Branch (then the Simulation & Analysis Group). The initial prototype of the project was developed by Rowan University's Software Engineering, Graphics, and Visualization (SEGV) research group through a Collaborative Research and Development Agreement (CRDA) with the FAA [2]. Over the years many students have worked on the prototype version adding additional requirements determined by CA. This prototype was used primarily within the CA Branch and several other groups or organizations that were working directly with CA.

The prototype version of the software had the ability to display one or several conflicts (loss of required separation standards between two flights) retrieved from the CA databases. The user could view the geometry of the recorded aircraft track in four dimensions and had full control over the time replay. The aircraft tracks were highlighted when the flights entered into a conflict. This system was not designed to be flexible, and was only intended to show the geometry of a conflict.

In December 2010, it was decided that the prototype had run its course. New requirements were developed that would take many of the lessons learned from the prototype and apply them to a flexible environment that would allow nearly any aviation concept to be visualized. It was released in August 2012 under the name Flexible Flight Traffic Exploration Visualization 4D. It is currently available to the public free of charge as a partially open-source development platform[1].

Software Overview

The FliteViz4D software is a platform that was made extensible so that it can help to analyze many different concepts. The FliteViz4D software is split into two parts: core software and plugins. Both parts are needed in order for FliteViz4D to be useful software.

The core software is the part of FliteViz4D that contains all of the logic for setting up and controlling the program. It contains mostly background processes that are not directly seen by the user, but without which, the plugins could not function. Functionalities of the core software include 3D visualization setup and control, camera control algorithms, time management, and recording and playback functionality. The core software is released as an executable only, with documentation, but no source is provided.

The plugins are the parts of the software that perform most of the functions that are directly visible to the user. The three main plugin types are loaders, stats, and layers; a fourth type, tools, will be described later. Loaders are used for collecting data and other information from a file or database table. Stats are used to display some numerical or string information to the user. They may contain complex algorithms to determine that information or simply retrieve the information from data that has been collected by a loader. Layers are used to display visual information to the user in the visualization. They can also provide interaction capabilities that allow the user to use the keyboard or mouse to perform some function, such as displaying more information about an airport when it is selected.

There is one other type of plugin called a tool. The tool category does not have a single generic purpose like the other three categories do. Tools can perform virtually any function. One tool may provide an interface to duplicate and modify flight paths, whereas another tool may provide functionality to write some data to a file. It is also possible that a tool is a standalone application in itself, and the plugin simply provides a link from FliteViz4D to the other application.

All plugins are used by attaching them to the core software. The core software provides plugins with the ability to interact with the 3D visualization, statistical displays, time controls, and other plugins. All of the complex 3D visualization setup is performed within the core software, allowing layer plugins to focus only on the items they are displaying, without worrying about visualization setup or interactions.

Capabilities

There are several different features of FliteViz4D that make it a powerful tool to the analyst. The first and foremost is the 3D Visualization. The 3D visualization allows users to view and analyze air traffic concepts in three dimensions.

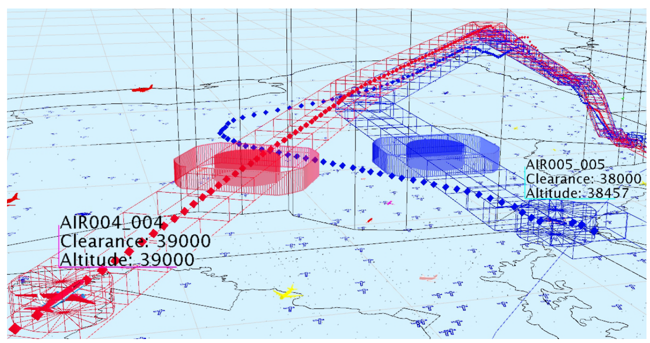

Figure 2. FliteViz4D displaying two trajectory predictions at a look-ahead time of 160 seconds

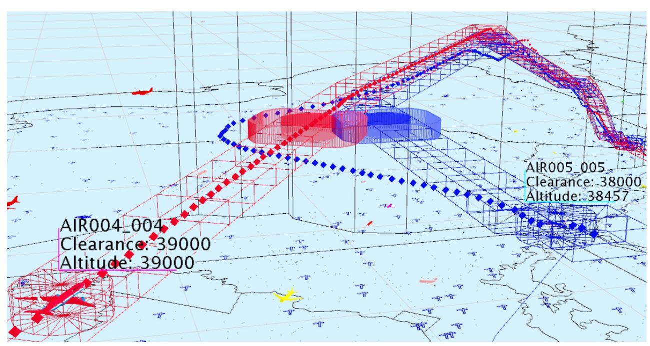

Figure 3. FliteViz4D displaying two trajectory predictions at a look-ahead time of 260 seconds

[1] http://acy.tc.faa.gov/fliteviz

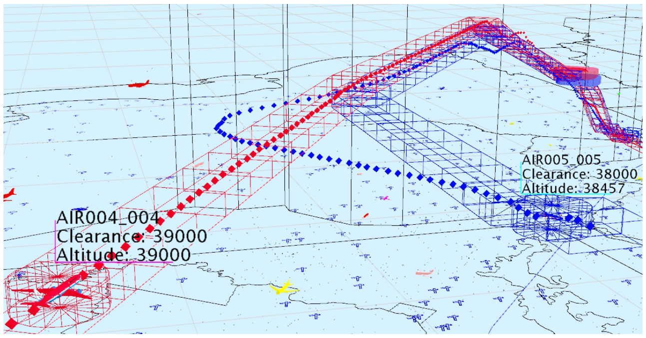

Figure 4. FliteViz4D displaying two trajectory predictions at a look-ahead time of 1200 seconds

A fourth dimension is added via a time control. The time feature has a slider similar to that in a video player. The user can drag the slider back and forth or allow the time to advance automatically at a speed as slow as real time and as fast as 1000x. As soon as the time advances, layers in the visualization are updated for the current time, which will cause some objects, such as flights, to animate as the time goes on.

Another temporal dimension is available to the user, which is called *look-ahead time*. Look-ahead time determines how far into the future the user wishes to look-ahead. In air traffic data, along with the events that are occurring now, there is often information that is predicting what will happen in the future. An example of this is a 4D Trajectory Prediction. The trajectory prediction is generated at a point in time, and it will appear at this point in time in FliteViz4D. It then looks forward by some amount of time, creating a predicted path into the future. Using the appropriate layer, the user can evaluate the trajectory prediction by sliding the look-ahead time slider (Figure 2, Figure 3, and Figure 4). The layers that make use of this future time will display to the user what the events would look like at that future time, given that the situation occurred exactly how it was predicted to occur. This look-ahead time capability is a very powerful tool that allows the user to peer into the future.

While visualization is a great tool for providing context and geometrical information, often there is more information required than visualization can convey to the user. In this situation, FliteViz has a statistics capability that can provide string or numerical information about a particular flight or a pair of flights. These plugins are referred to as stats. Like the visualization, stats update whenever the current time or look-ahead time changes. This allows stats to tell the user the current state or future state of the flight or pair of flights. Stats can contain anything from the departure airport or aircraft type, which do not change at all and are simply retrieved from collected information, to the longitudinal component of the trajectory error, which is calculated every time the current time or look-ahead time is updated. All the statistics are shown in the *Flight Details* window that is shown middle-left in Figure 1. There also may be some layers that make use of these statistics, such as the *Label* layer, which is described in the next section.

3D Visualization

The most prominent feature of FliteViz4D is the ability to display information in a 3D environment. The core software sets up the 3D environment and handles camera movements and interactions, but does not actually display any information in the visualization. The plugins enabled by the user determine what information is displayed in the 3D environment, which gives almost limitless possibilities to what information can be displayed in the visualization.

In the case of Figure 5, the visualization is displaying flight paths, trajectory predictions, routes, center boundaries, state lines, and airports, among other things. The current location of the flight is indicated by the 3D model of the airplane, while the diamonds protruding behind and in front of the two aircraft show the previous and future locations of the flights.

The wireframe cylinder surrounding the aircraft represents the minimum separation requirements in en route airspace (5 nmi. x 1000 ft.). The thick translucent lines near the aircraft are the trajectory predictions, which represent where some decision support software predicts that the flights will fly in the future. In Figure 5, for the red flight it can be seen that the trajectory prediction was not very accurate, since the thick translucent line (the prediction) does not follow the diamonds, which we know to be the actual future position of the aircraft.

This scenario is using pre-recorded flight data, so FliteViz4D already has all the track data available. FliteViz4D is also capable of receiving data while running (e.g. live feed data), so it is not always the case that it knows the entire flight path.

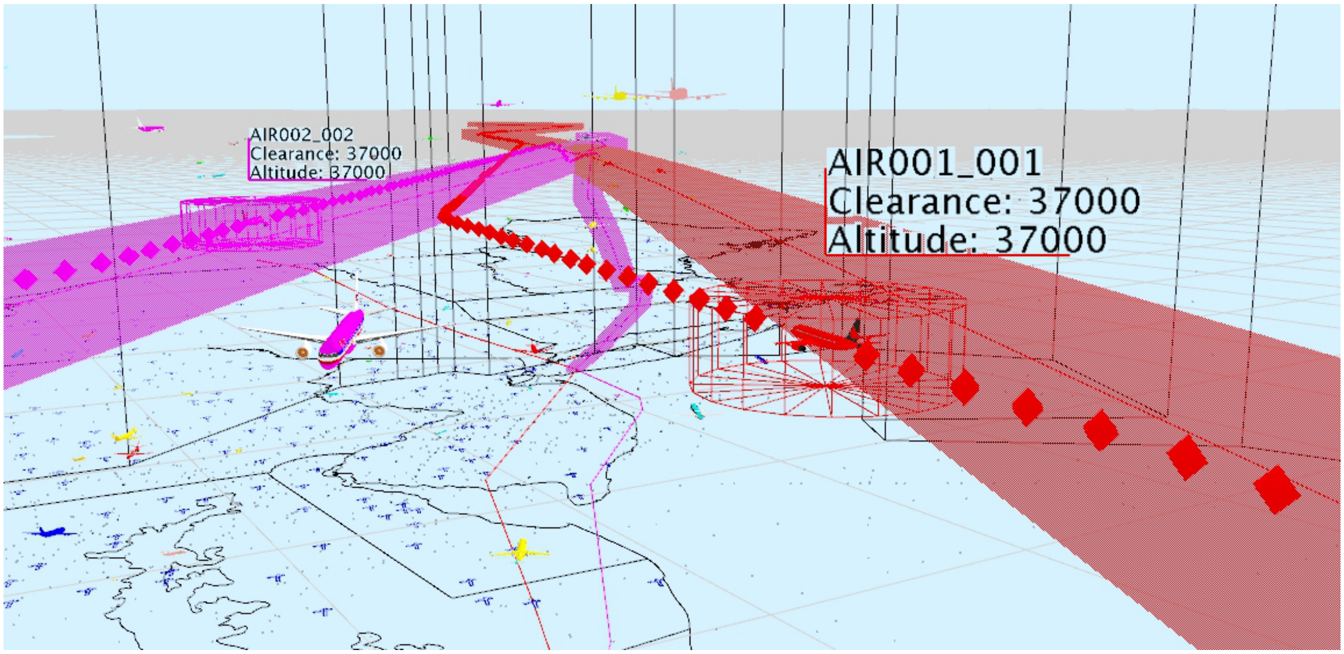

Figure 5. FliteViz4D's 3D Visualization displaying flights and trajectory predictions

The labels are connected to the current location of the flight and display the aircraft ID as well as two stats that can be selected by the user. In this case, they are showing clearance altitude and actual altitude. Finally, the routes (flight plans) are the thin lines drawn on the ground.

The colors are selected randomly from a set of six colors. It uses the idea of color by association, in which everything associated with a particular flight will display with its primary color matching the primary color of the aircraft, so that the user's eye quickly associates them. Six different colors are used to provide a good possibility of two flights of interest being different colors, without the risk of overwhelming the user with so many colors. If flights of interest do happen to be the same color, the user may select the flight and modify the color to any color in the RGB color space.

In this particular visualization, only two flights have a lot of information displayed for them, even though there are many other flights in the visualization. This is a result of the user having the additional layers hidden for the entire scenario. Then, through the use of another plugin, the user is able to select the two flights of interest and display the hidden layers for only the selected flights. This allows the user to focus on flights of interest without the visualization being overcrowded by the many other flights, yet the often important context of the other flights in the airspace is not lost.

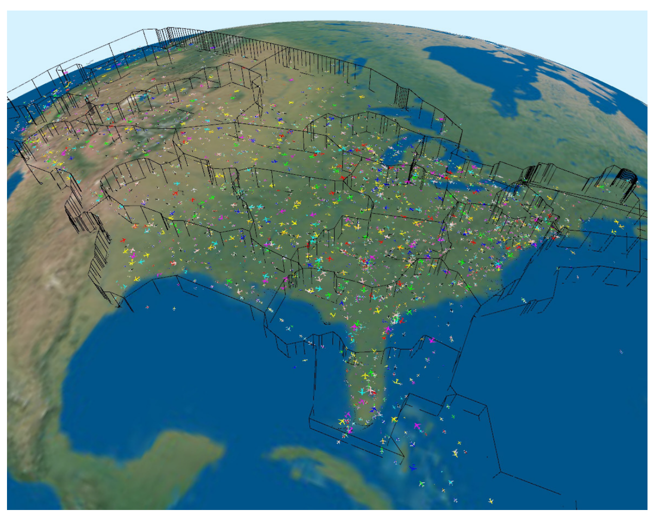

Figure 6. FliteViz4D displaying over 5000 flights across the Continental US on a globe visualization

There are also some layers in this visualization that are not directly associated with flights. The vertical lines in the background are displaying the boundary of the air route traffic control center (ARTCC). The lines on the ground display state boundaries to provide the user with a spatial context. The blue icons on the ground display locations of airports. The pink lines on the ground are graticule lines that are drawn every 1° latitude and longitude allowing the user to see the shape of the map projection being used.

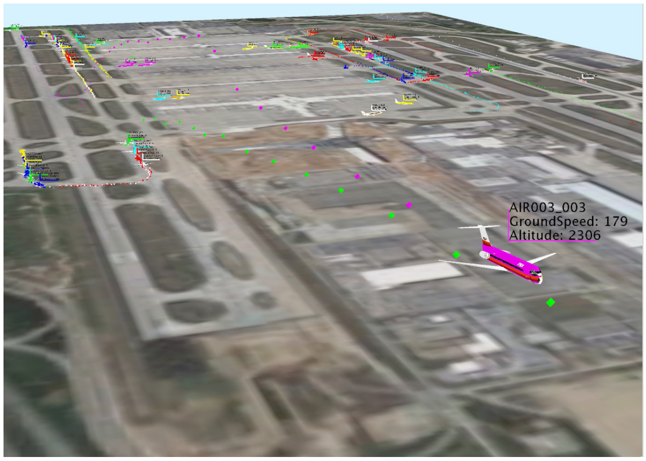

Figure 7. FliteViz4D displaying an airport ground texture and ASDE-X (ground) flight data

Everything displayed in the visualization is a separate plugin that can be enabled or disabled at will. Even at just the current flight position, the airplane model, the wireframe cylinder, and the label are each separate plugins. Typically, everything that should logically be a separate plugin is made into a separate plugin to promote decoupling. Changing the enabled plugins can completely change how FliteViz4D displays, allowing it to be used in many different situations as can be seen in Figure 6, Figure 7, and Figure 8.

In Figure 6 FliteViz4D is being used to display flights in the entire NAS. The user has chosen to view the data on a globe visualization, rather than a flattened map, as in most other figures. At the point of the screen capture, over 5000 flights were being displayed. A total of 50,019 flights were loaded into this scenario.

Figure 7 is displaying ground data and low-altitude flight data at Hartsfield-Jackson Atlanta International Airport (KATL). This figure, along with Figure 6, is a prime example of the scalability of FliteViz4D

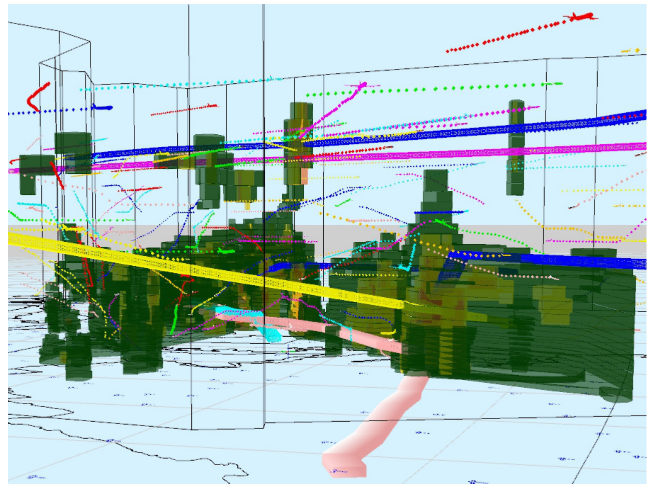

Figure 8. Side view of 4D Convective weather polygons and flight reroutes in FliteViz.

Figure 8 is displaying 4D convective weather information, along with flights and trajectories. The large dark green areas in the visualization are areas of convective activity. Dark green indicates an area of moderate convective activity. Inside most of the dark green areas, there are also yellow, orange, and red areas that indicate increasingly severe convective activity. The drawings of the convective weather in the visualization are translucent so that more severe activity can be seen through the less severe activity. An example of this can be seen in the figure as smaller yellow and orange spots can be seen inside the green areas. In this figure, light convective activity (normally indicated by light green) has been hidden to allow easier viewing of the more severe activity.

3D Camera

Another important feature of FliteViz4D is its 3D Camera. Any 3D visualization is only as powerful as it is easy to navigate. With three dimensions being displayed on a two-dimensional screen it is very important for the user to be able to quickly and easily navigate around the visualization to gather all the information available. Looking at the visualization from a single point-of-view may result in the user not seeing some important information. Many different 3D cameras have been developed for use in other 3D visualizations, but FliteViz4D has some unique requirements that these cameras did not meet.

Google Earth[2] contains one example of an easy-to-use 3D camera. Google Earth's camera was built specifically for viewing the ground via satellite imagery, and it is very effective at doing so. However, FliteViz4D requires the ability to look at the ground as well as focus on any other altitude that flights may be at. The agility and precision of the camera in a visualization needs to be appropriately tuned for the scale and detail allowed by the visualization, e.g. minute detail is lost to the user if the camera is unable to zoom in close enough to see it. This is complicated to accomplish while retaining ease of use in a visualization designed with variable levels of detail and arbitrary focus. FliteViz4D may contain a lot of information spread throughout the visualization while the user wishes to focus on a single small part of the whole data set, e.g. one airport in a NAS dataset.

This requirement is very similar to a three-dimensional video game where, though the visualization is very large, the user is only interested in the player's current position within the visualization. Taking concepts from 3D video games and those used within Google Earth and other similar globe visualizations, CA developed a powerful camera that meets all requirements of a visualization tool such as FliteViz4D.

The main feature of the camera that sets it apart from many others is the focal point. The focal point of the camera is where the user is currently focused. In many 3D cameras this focal point is either immovable or only movable in specific directions. In visualization like Google Earth, the focal point is only movable along the surface of the earth. In FliteViz4D this focal point may be moved along all three axes, allowing the user to focus on any point in space.

The next features required by the camera are the abilities to view the focal point from any angle or distance. Viewing from any angle is accomplished by providing the ability to rotate side-to-side as well as up-and-down providing a full 3D range of motion. Viewing from any distance is accomplished by providing the capability to zoom in and out. As the user zooms in closer to the focal point, movements become more precise allowing the camera to be just as easy to navigate from one end of a runway to the other as it is to navigate from one end of the country to the other.

[2] http://www.google.com/earth

The full 3D range of motion of the focal point allows it to be used in many more ways than simply navigating around the visualization. One example is that the focal point can be attached to a flight of interest, so that the camera will then follow that flight as it moves.

The last requirement of the 3D camera is for it to be easy to use. All of the navigation of the camera can be performed using a standard mouse that includes a wheel. A touchpad or trackball mouse can be used, but it is slightly more difficult. There is a learning curve, as there will be with any 3D visualization, but in a user survey performed (detailed later), it has been found to be minimal.

Live Feed DVR

Many are familiar with the concept of a Digital Video Recorder (DVR) attached to a cable television. The DVR allows the user to pause and rewind a live television feed. FliteViz4D makes use of this same concept, allowing the user to pause and rewind a live feed to allow closer observation of some event. Live feeds in FliteViz4D can be sent through a network socket connection using a very simple protocol. The live feed may include flight paths, trajectories, routes, clearances, and time, and may come from any source. The source may be actual live flight traffic, or it may simply be from a simulation platform. Therefore, this data can be sent real time or fast time.

The DVR system provides additional controls to the user that can jump to the current live feed or give time control to the user, allowing them to move the time to any point in the past. When the live feed time is selected, the time will update whenever a new message is received that contains a time stamp. Some of these messages may be track reports and others may be only time updates. The user may select which types of messages are allowed to control the time.

Multiple Scenario Visualization

Another major feature of FliteViz4D is its ability to display multiple scenarios of air traffic. While the definition of "scenario" is dependent upon the data type, typically a scenario is a single set of flights over a period of time. The flights may cover a very small area, such as an airport, or a large area, like a continent or even the entire world. FliteViz4D allows multiples of these to be loaded into the visualization at one time. This provides an advantage with smaller scenarios in that multiples can be combined together to view as if they are one larger scenario. For example, if data was collected in two centers for the same timeframe, they can be displayed side-by-side without having to merge the underlying datasets.

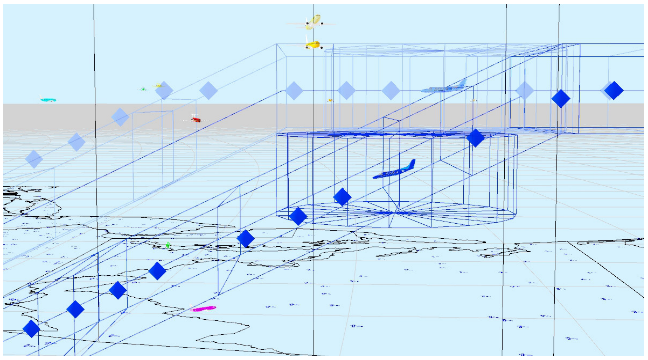

Figure 9. Multiple scenarios being compared in FliteViz4D by "ghosting" one of the scenarios

Another advantage of this is the ability to compare scenarios. The multiple scenarios may actually be all the same flight data, but each scenario comes from a different source. For instance, one scenario may be recorded air traffic, and the other scenario is air traffic that was generated using a fast-time simulator on the flight plans of the flights from the recording. Loading these two scenarios into the same visualization would overlay one scenario on top of the other. The user can then choose to "ghost" one of the scenarios, so it is easy to view the differences between the two scenarios. In Figure 9, two scenarios are being displayed, both from human-in-the-loop simulations. Both scenarios contain the same flights, but each time they were flown different ways due to changes in the decision support tools provided to the controllers in the study. The first scenario is solid in the visualization, representing the baseline that the "ghost" is being compared against. The second scenario is ghosted, so it appears transparent and lighter than the other scenario. This way it can easily be distinguished from the other scenario.

In this example it can be seen that in the one scenario, the flight begins to descend earlier than in the ghosted scenario. In this particular case, descending the flight earlier caused this flight to come much closer to a loss of separation than it did in the ghosted scenario.

Use Cases

FliteViz4D has been designed to have many uses in many different studies and other situations. Descriptions of some of these uses follow.

Past Uses

FliteViz4D has already been used many times in many different studies. It is very common for it to be used briefly in a study to view geometry of a flight or even just to present a concept or idea to others. There are several studies, however, in which FliteViz4D has played a major role in the study's success. The following two examples were two major studies that FliteViz4D played a large part in.

Weather Reroute Analysis

The weather reroute analysis was a study performed by the FAA in 2011 [3]. At the time, the initial pre-release version of FliteViz4D had just been developed, so the study presented an opportunity to determine if the goals of flexibility and ease of use of the new software had been met.

The tasks of the study were to determine how close flights flew to convective weather and how far they would reroute around the weather. The first task had been accomplished before by MIT Lincoln Labs (LL), so the study performed by the FAA was simply to determine if anything had changed in the few years since the LL study with the additional weather related technology in the cockpits. The second task, however, had been attempted by previous studies but had never provided satisfactory results.

The issue with determining how far aircraft reroute around the weather is first determining when they are rerouting due to weather. One way to do this is to look at weather, and look at flights and pick out flights that seem to be rerouting around the weather. This is tedious and can only provide a very small data set in a reasonable amount of time. The problem needed to be solved algorithmically, but more information was needed before an algorithm could be developed.

The convective weather retrieved by the FAA is a 4-dimensional grid that updates every 2.5 minutes and has a measurement of reflectivity level at every 2 nmi. in the horizontal plane at 31 altitude bands. This results in over 760 million data points every 2.5 minutes and approximately 438 billion data points in a 24-hour period for the continental United States. This is far too much data to process for flight reroutes, and the grid format does not provide an easy way to algorithmically determine if a flight is inside a convective cell or not.

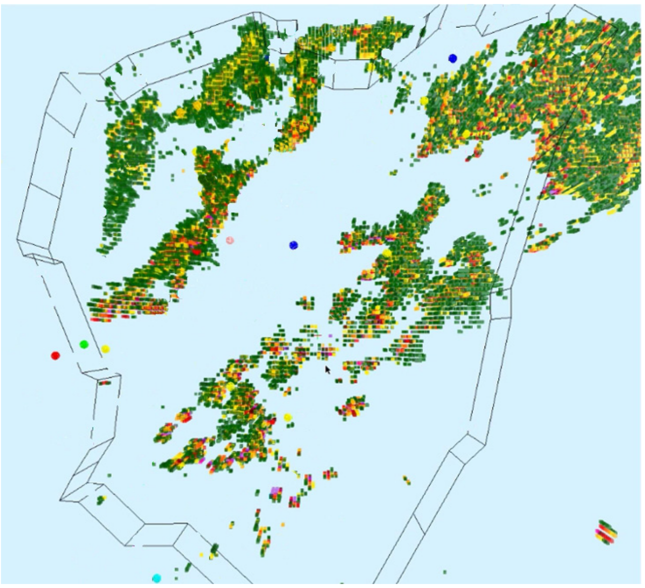

Figure 10. Top view of convective weather grid in the Washington ARTCC displayed in FliteViz4D

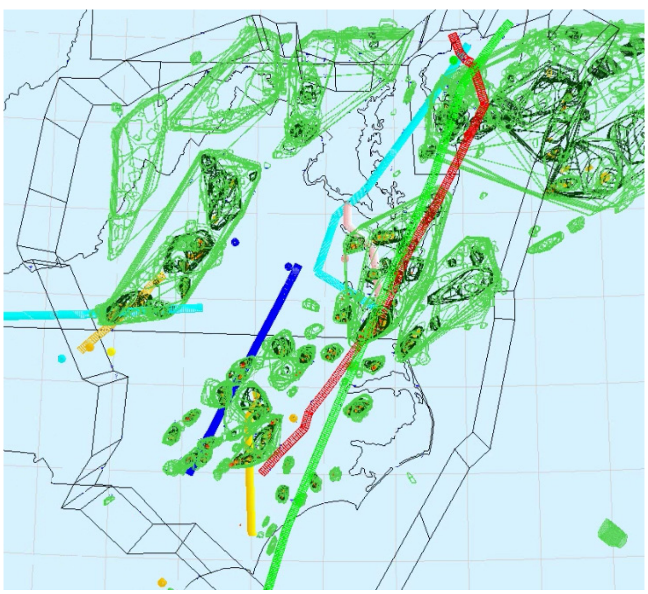

Figure 11. Top view of convective weather polygons generated by the FAA's algorithm

To generate more useable information from the convective weather grid, the FAA developed a polygon generation algorithm. The polygons have the benefits of being light-weight in comparison with the raw data, so are easier to store. Also, since they have a shape and a volume, it is much easier to determine algorithmically if a flight is inside a weather cell. While weather polygon generation is not a new concept, the specific use for the polygons required some modification to previous algorithms used. In order to ensure that the modified algorithm provided good results, the gridded convective weather and the polygons were both added into FliteViz4D as layer plugins (Figure 10 and Figure 11).

Overlaying the two layers, the analyst could verify that the output polygons provided a good representation of the actual convective weather. This was done in several iterations, tuning the algorithm's input properties each time before the proper settings were found. FliteViz4D played a major role in this part of the study, since each layer took only several hours to create, whereas it would have taken much more time to develop a new tool to visualize the grid and polygons together. This is also a prime example of using FliteViz4D in an unusual manner that does not involve flight information. Only the weather information was selected for display in this analysis.

The next step in the study was to actually develop an algorithm that could determine when a flight is rerouting around weather. Past studies have assumed that when there is convective activity present in the airspace, any flight that is diverted from its current flight plan must be rerouting due to weather. However, there are many reasons a flight may be diverted, and these occur in clear airspace as well. Flights may be diverted due to traffic, the pilot may request to proceed directly to a portion of the flight plan further down, rather than making a wide turn, or the controller may vector the flight to line it up behind or in front of other flights for descent into an airport. Unfortunately, there is no flag that exists in any recorded data that tells whether the flight was rerouted because of weather.

The first step in accomplishing this task was to gather information about what flights are doing to reroute around the weather. To gather this information, the flights and weather polygons were loaded into FliteViz4D and observed. Several important discoveries were made at this point. First, it was determined that flights did not always vector off of their flight plan in order to avoid weather, but sometimes would receive a flight plan amendment that would take them around the convective activity. It was also noticed that flights would fly relatively close to the convective activity, both in front of it and behind it. However, when the flights do reroute around convective activity, in most cases immediately before the reroute begins, the current flight plan has the flight proceeding directly through severe convective activity.

These findings, made through FliteViz4D, allowed the FAA to use prior experience in trajectory prediction to generate the predicted location of where flights would be if they had not rerouted. Using these predicted positions, an algorithm could be used to determine if the reroute was indeed due to convective activity. The results of the algorithm were later validated using FliteViz4D once again, and several outliers that had very large reroute distances were also analyzed to determine why they rerouted so far.

FliteViz4D played a major role in the success of this study, and it is even possible that without the ability to easily add the visualizations into FliteViz4D, the FAA may not have been able to accomplish the goals set by the required date.

Separation Management Modern Procedures

Separation Management Modern Procedures is a large project with many different studies and development being done simultaneously by different groups within the FAA and support contractors. The Concept Analysis Branch is tasked with determining recommendations for modifications to be made to the current conflict probe decision support tool within En Route Automation Modernization (ERAM) [4] [5]. This task is in support of improving the conflict probe so it can be used by the air traffic controller on the radar side (R-side).

The prototype of the FlightGUI project was initially developed to support analyses for the separation management project [1]. The prototype and the current version of FliteViz4D have been used many times in many different analyses related to separation management.

A conflict probe uses 4-dimensional trajectory predictions to predict where the aircraft will be located at in the future, up to 20 minutes ahead. It then searches these trajectories for potential losses of separation. A loss of separation occurs when two aircraft come closer than the required separation standards that have been set forth by the FAA (currently 5 nmi. horizontally by 1000 ft. vertically in most en route airspace). The purpose of a conflict probe is to alert the ATC to potential future losses of separation with enough warning time that the controller can act upon the alert with strategic planning to avoid the loss of separation without causing additional issues.

Because the trajectories are predicting the future, it is impossible for them to be accurate every time. In order to account for the potential inaccuracy, buffer zones are used around the predicted positions of two flights that are dependent upon the geometry of the two flights in relation to each other.

There are many variables present in the conflict probe system. So, every time something is modified, the system must be analyzed in detail to determine what effects that modification has. Statistical software is used initially to determine the high level effects. After that step, FliteViz4D plays a large role in determining how the modification has affected particular situations that provide different results now. During this analysis stage, the flight path, route, clearance altitudes, trajectory prediction, buffer zones, and even airspace are all important factors in determining the effects of the modifications. All of these things can be visualized in FliteViz4D, including the future position.

Other Example Uses

There has been much discussion with many different groups internal to the FAA, and a few external to the FAA, about possible future uses for FliteViz4D. There are many different air traffic concepts that would benefit from being visualized in FliteViz4D for analysis and requirements determination, but there are possibilities beyond the concept analysis realm that FliteViz4D would be useful for.

Using the Live Feed DVR, FliteViz4D could be used as a monitoring system, allowing the user to view the current state of live air traffic, a fast time simulation, or even a human-in-the-loop simulation from a remote location. However, though the software can be used to view the state of live air traffic, it is intended for use as an analysis and informational tool only, not for air traffic control purposes.

Software Design

The design of the FliteViz4D software is a very important aspect of the software. It allows the software to be easy to extend by any developer and easy to use by any user. It is a general rule of software engineering that software should be modular, cohesive, and low in coupling. This means that each function of the software should be well defined and as independent as possible from other functions of the software. For instance, the air traffic data should be contained in a single location and stored independent of how it is used. By designing the software like this, it allows functionality to be added and removed without causing other functionality to break and without needing any knowledge of how the other functionalities work.

In FliteViz4D this was accomplished by separating the visualization and statistics modules from the data collection and storage modules. Additional data collection and storage functionality can be added independently of the visualization or statistics functionalities, and vice versa. As a developer, the only requirement is to have knowledge of the interfaces the plugins must interact with. As a user, adding the plugin is as simple as installing it using a simple graphical interface, then selecting the plugin or group of plugins from a list within FliteViz4D.

Developing a Plugin

While FliteViz4D already has many different plugins available for reading data from many different sources and for displaying many different types of information to the user, another main feature of FliteViz4D is the ease at which additional plugins can be added. Often, organizations will have a need to evaluate new information in FliteViz4D that has not been observed previously. For this situation, developers are provided with an extensive and well-documented interface for interacting with the features of FliteViz4D.

FliteViz4D is built entirely in Java 6 using Java Bindings for Open GL (JOGL) 1.1.1 for the 3D graphics. Developers who have a working knowledge of Java should have no trouble developing loader or stat plugins, but some basic knowledge of OpenGL may be needed to develop layers.

There are many resources available online for OpenGL development, many of which provide much more information than is needed for FliteViz4D. When developing a layer in FliteViz4D, most of the difficult work has already been completed, such as setting up the 3D environment, establishing the lighting, and creating a moveable camera. As a result, creating a layer in FliteViz4D can be as simple as one or two calls to the JOGL library. Most of the knowledge required for creating a layer can even be gained by studying other open-source layer plugins available on the FliteViz4D website.

The advantage of the plugins being open source is that they provide many different references for developing additional plugins. As more plugins are added to the website, developers are more likely to find one that already does most of the work they need to perform. Since they are all open source, a developer may modify it in any way desired, as long as it is used within the terms of the open source license agreement.

User Survey

The user base for FliteViz4D currently only extends to about 20 people. About one third of those are external contractors working with the FAA or other federal agencies. The rest of the users are internal to the FAA, spread across three different groups. These users were given a survey to determine what their views of FliteViz4D are.

The survey was taken anonymously. It contains a total of 14 questions. The first three questions focus on how much experience the user has with FliteViz4D. The next seven questions are five-point Likert questions ranging from “Completely Disagree” to “Completely Agree” gauging user opinion of the effectiveness or ineffectiveness of FliteViz4D features. The next question asks for a rating from 1 to 10 for how useful FliteViz4D has been to the user for their particular work. The last three questions are open-ended questions that request a description of the work done with FliteViz4D, suggestions for improving FliteViz4D, and any additional comments the user wanted to respond with.

A total of eight responses were received from the users. Experience ranged from using it weekly for the past year to using it once a month for the past 3 months.

The seven Likert items were converted to a five-point Likert scale measuring from 1 to 5, with “Completely Disagree" being equal to 1 and "Completely Agree" being equal to 5. All seven questions had responses with a standard deviation of less than 1 point, which shows that the users generally agreed on all of the questions.

Two questions were geared toward FliteViz in general. The results of these two questions were that FliteViz does not have too steep of a learning curve, and it is very easy to use.

Two questions focused on the 3D visualization aspect in particular. The results of these two questions were that the visualization is very easy to understand and depicts information in a useful way.

Three questions were asked about the 3D Camera. Users were somewhat undecided on if it had a steep learning curve, though they leaned more toward it not being too steep. It was determined to be mostly easy to use and provides the capabilities required of a camera for this type of visualization.

It was given an average rating of 9.3 out of 10 with a standard deviation of 0.95 to the question about how useful the software is to the user.

Some of the uses listed by the users were trajectory predictor analysis, visualization of traffic for Staffed NextGen Towers, viewing convective weather data, evaluating map conversions, conflict probe analysis, scenario comparisons, and analyzing merged ASDE-X and PDARS data.

Future Work & Possibilities

Though FliteViz is currently well established after several past iterations, there is always more work to perform. Some of this work is on FliteViz whereas other work is on supporting tools.

The website set up currently is a temporary website with a very simple interface that provides access to FliteViz and the open source plugins that have been developed by Concept Analysis and other groups within the FAA that work closely with CA. In the future, an interface will be provided for other developers to upload their own plugins for FliteViz. The hope is to establish a community, sharing software tools among each other, with the common goal of bringing the aviation industry into the next generation.

FliteViz is constantly being upgraded by CA. Some of these upgrades may be simply additional plugins, whereas others are upgrades on the core software, either fixing bugs or adding additional features.

Conclusion

FliteViz is a flexible, easy to use, 4D visualization environment for evaluating air traffic concepts. Its ability to use plugins makes it a powerful tool to anyone in the aviation industry, due to the flexibility and ease of use for developers and the analyst alike. A user study was performed on a small set of users that found it has accomplished the goals it was designed for. The software has already been used in many past studies, including a weather reroute analysis and conflict probe analysis, and is currently being used in many others by the FAA and supporting contractors. It has been released to the public as a combination of open source and proprietary software, all of which is available free of charge to the aviation community[3].

References

[1] Santiago, Confesor, Mike Paglione, Robert Oaks, Dr. Adrian Rusu, 2006, Flight Graphical User Interface: A Visualization Application for Analyzing Flight Conflict Probe Tools, 51st Air Traffic Control Association Annual Conference Proceedings.

[2] Santiago, Confesor, Chu Yao, Andrew Crowell, Dr. Adrian Rusu, 2008, Government and Academia Partnership to Test and Evaluate Air Traffic Control Decision Support Software, International Testing and Evaluation Association (ITEA) Journal, Volume 30(1), pp. 31-38.

[3] Young, Jessica, Andrew Crowell, Andrew Fabian, Albert Schwartz, Dan DiBuccio, 2011, A Measure of Efficiency of Current Flight Operations in Convective Weather and Use of Weather Polygons in Fast-Time Simulation, DOT/FAA/TC-TN11/9.

[4] Crowell, Andrew, Andrew Fabian, Christina Young, Ph. D, Ben Musialek, Mike Paglione, 2011, Evaluation of Parameter Adjustments to the En Route Automation Modernization's Conflict Probe, DOT/FAA/TC-TN12/2.

[3] http://acy.tc.faa.gov/fliteviz

[5] Crowell, Andrew, Andrew Fabian, Christina Young, Ph. D, Ben Musialek, Mike Paglione, 2011, Evaluation of Prototype Enhancements to the En Route Automation Modernization's Conflict Probe, DOT/FAA/TC-TN12/3.

31st Digital Avionics Systems Conference
October 14-18, 2012