

User Centered, Application Independent Visualization of National Airspace Data

James R. Murphy*

NASA Ames Research Center MS 243-1, Moffett Field, CA, 94035

and

Susan Hinton†

AerospaceComputing Inc., Mountain View, CA, 94043

This paper describes an application independent software tool, IV4D, built to visualize animated and still 3D National Airspace System (NAS) data specifically for aeronautics engineers who research aggregate, as well as single, flight efficiencies and behavior. IV4D was originally developed in a joint effort between the National Aeronautics and Space Administration (NASA) and the Air Force Research Laboratory (AFRL) to support the visualization of air traffic data from the Airspace Concept Evaluation System (ACES) simulation program. The three main challenges tackled by IV4D developers were: 1) determining how to distill multiple NASA data formats into a few minimal dataset types; 2) creating an environment, consisting of a user interface, heuristic algorithms, and retained metadata, that facilitates easy setup and fast visualization; and 3) maximizing the user's ability to utilize the extended range of visualization available with AFRL's existing 3D technologies. IV4D is currently being used by air traffic management researchers at NASA's Ames and Langley Research Centers to support data visualizations.

I. Introduction

The question of how data can best be visualized to help guide research is an aspect of many air traffic management (ATM) research projects that tends to be postponed or overlooked altogether. Often the cost of developing a data visualization tool, both in terms of support hours and price, is considered prohibitive, and effective visualization tools are included only for larger, more expensive software efforts that require the development of an interactive display for other reasons. Research efforts without a significant display component often visualize data using third party spreadsheets or statistical packages. Examples of large, well-funded, multi-year software development projects in support of ATM research at the National Aeronautics and Space Administration (NASA) Ames Research Center include the Center TRACON Automation System (CTAS) and the Future ATM Concepts Evaluation Tool (FACET). CTAS is a suite of ATM decision support tools developed to investigate concepts of orderly flow of aircraft into and

* Software Manager, Aviation Systems Division, Mail Stop: 243-1, AIAA senior member.

† Software Lead, ACI, Aviation Systems Division, Mail Stop: 210-08, AIAA member.

through en route and terminal airspace.^{1,2,3} The CTAS displays were, therefore, developed primarily for simulated and operational air traffic control use rather than for general visualization.^{2,4} FACET, on the other hand, was built to examine the efficiency of Traffic Flow Management (TFM) algorithms, for moving flights through the National Airspace System (NAS) and visualization is an important component in assessing results.⁵ Though improving on the utilities of CTAS, FACET employs older Java Native Interface (JNI) software technology as its basis for a visualization display, and requires users of external applications to issue application programming interface (API) draw commands to display data.⁶ Though the displays for both of these toolsets are sufficient for the needs of their respective users, augmenting the displays to visualize new and different types of data is cumbersome and usually requires a significant development effort.

The Airspace Concept Evaluation System (ACES) simulation application was developed, under the Virtual Airspace Modeling and Simulation (VAMS) project, to simulate NAS operations.⁷ The initial development of ACES focused on modeling all aspects of aircraft flight from gate-to-gate.⁸ Though development efforts included a visualizing component, ACES simulations can take hours to run, and researchers wanted a way to visualize data independent of simulation runs. To provide researchers with an independent way to visualize ACES data, NASA Ames engineers collaborated with engineers from the Air Force Research Laboratory (AFRL), who have extensive experience with 3D technologies and complex data visualization and had a desire to extend their existing 3D visualization software to a wider domain.^{9,10} The result of this joint effort was the Interactive Visualization of NAS Data in 4 Dimensions (IV4D) software that combined the visualization capabilities of the AFRL technologies and the lessons learned from NASA's previous ATM data display efforts. The development of IV4D was driven by two primary design philosophies: 1) the visualization software should not be tied to a specific software application; and 2) the interface to the data and visualization algorithms should be tailored to the needs of a user with air traffic domain knowledge but with little experience in software development.

This paper documents the main design features and decision points in developing the IV4D, including the partnership with AFRL and the use of their existing technologies. The intent is to educate the research community on the IV4D software and to illustrate its usefulness for research.

II. Background

The terms data display and data visualization are often used interchangeably, but there is a distinction; in this paper "display" refers to any depiction of data to the user, regardless of its form, while "visualization" refers to a graphical display of data that promotes spatial and temporal awareness. For example, the set of all airport latitude/longitude locations can be shown to a user in a text editor or shown graphically with each location plotted on a map. Both are displays of the data, but only the second is defined as a visualization. Fig. 1 shows an example of an ATM visualization.

IV4D provides researchers with the capability to generate and display visualizations of ATM data using a simple but powerful user interface. The IV4D design requirements were:

1. determine data formats to visualize;
2. create an easy to learn user interface to specify and extract data to visualize;

3. develop heuristics for determining appropriate visualization components selected by the user; and
4. allow a user to easily apply the settings from an existing visualization to new data and allow a user to combine visualizations together to form a new visualization.

As the ACES program was developed, it became evident that a method to help visualize input and output data was necessary. The requirement was for an easy to understand, easy to use, and reliable tool that researchers could use to study and understand the NAS using data from varied sources (including ACES). However, the form the visualization tool should take was not immediately apparent. To determine a technical approach for developing the tool, NASA examined existing solutions and tools including open source and external software tools such as Google Earth¹¹, focusing specifically on user interface solutions and visualization technologies useful for displaying aeronautical simulation data. The outcome was a decision to develop a high performing, application/platform independent, visualization tool more similar to Google Earth than to CTAS or FACET.

There were two major challenges to creating an application independent visualization for the NAS. First, the visualizing application required an accurate and appealing visualization capable of simultaneously animating thousands of flights and airspace polygons, such as airspace boundaries and weather. Second, it needed to directly read many different types of data, or have a standard format to which external data could be easily transformed. To these a third high-level challenge was added: to make the tool as easy to use as possible since users, in general, avoid software that involves learning many new and complex procedures.

To achieve the specified IV4D design requirements, NASA partnered with AFRL and developed the layered approach shown in Fig. 2. IV4D was conceived as the top layer (Fig. 2), building upon existing software code layers, each with their own sets of interfaces. Starting with the bottom layer, each successive layer consolidates functionality of the layer immediately below leading to more a specialized design for visualizing objects in the NAS.

A. JView

The lowest specialized layer of software upon which IV4D is built (see Fig 2) consists of the Air Force graphics engine called JView.⁹ AFRL has worked on this software since the mid-1990s when they needed to animate and visualize their data in 3D on multiple computing platforms. As the nature of the data was not always well defined or known a priori, the software had to allow for changing data formats, such as units changing from feet to meters. JView was,



Figure 1. Visualization of ATM data. Visualization created by NASA researchers to show ZID and ZTL Center airspace reconfigured after applying two separate mathematically-based algorithms to flight track data through each respective Center.

therefore, built using two widely adopted open source code layers that were rapidly evolving: Java, and a graphics layer on top of Java, known as JOGL, or Java OpenGL (see Fig. 2).⁹ (JOGL is being incorporated into the Java language, starting with Java version 6.)

JView is a high-level graphics engine that supports simulation and modeling and is not an application for end users. Instead, it contains a toolset for software developers who write data visualization applications. JView was created to facilitate rapid development of animated and still visualizations that are highly accurate, 3D platform independent, and capable of using multiple coordinate systems such as Cartesian X-Y-Z or Geodetic Latitude/Longitude and Height (or altitude). JView provides the basic 3D canvas upon which IV4D visualizations are displayed (Fig. 2).

B. NAS Viewer

The NAS Viewer (known originally as the ACES Viewer) uses advanced 3D techniques to visualize ATM data. It is built on the base JView technologies to provide users with an interface to JView visualization capabilities, along with a mechanism for configuring and constructing a 3D Scene or 3D picture of data (shown as the green mid-tier boxes in Fig. 2).⁹ For example, Fig. 3 illustrates the use of one of the preprogrammed visualizations to depict flight density as represented by regions of colored metaballs[‡]. Though geared toward air traffic and airspace data, the tool is disassociated from the content by providing an abstracted interface to the data. This allows flexibility in the type and content of the data that can be displayed. Building the data scenes, however, can be cumbersome.

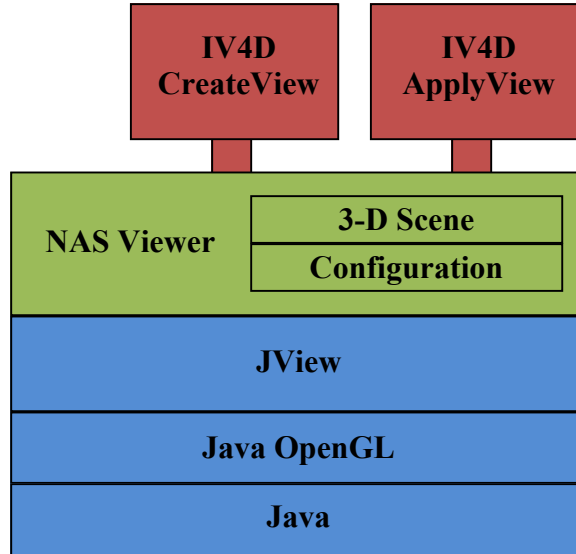


Figure 2. Architectural View of IV4D. *IV4D plugs into the NAS Viewer, which consolidates JView functionalities for the purpose of viewing the NAS.*

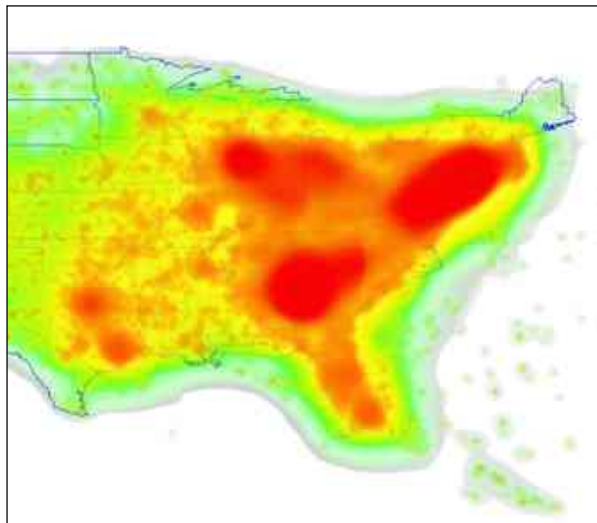


Figure 3. The NAS Viewer’s 3D Scene. *One of several preprogrammed visualizations that come with the NAS Viewer, showing increasing flight density represented by regions of colored metaballs. Though appealing, the preprogrammed visualizations do little to advance NASA research goals.*

[‡] ‘Metaballs’ is a standard term used by graphics programmers to describe organic-looking n-dimensional objects, which can be described by a specific set of mathematical algorithms.

The Composition panel in the NAS Viewer, shown in Fig. 4, is one of the mechanisms used to specify and configure the data that is to be displayed on the 3D Scene. On the left side of the panel the data, data sources, filters, and other data transforming mechanisms are represented as boxes that must be linked together in order to produce a desired result. Aspects of each datum, such as symbol, size, color, and/or labels are specified on the right side of the panel. This provides nearly complete flexibility in displaying data. Setting up and manipulating the data, however, requires support from expert users of the NAS Viewer toolset. (Additional examples of the Composition Graph can be seen in Figs. 17 – 20 in the Appendix.)

Building on the visualization functionalities embedded in JView and the data parsing and selection functionalities of the NAS Viewer, NASA and AFRL engineers developed the IV4D software that included the necessary flexibility to allow a novice user to visualize ATM data.

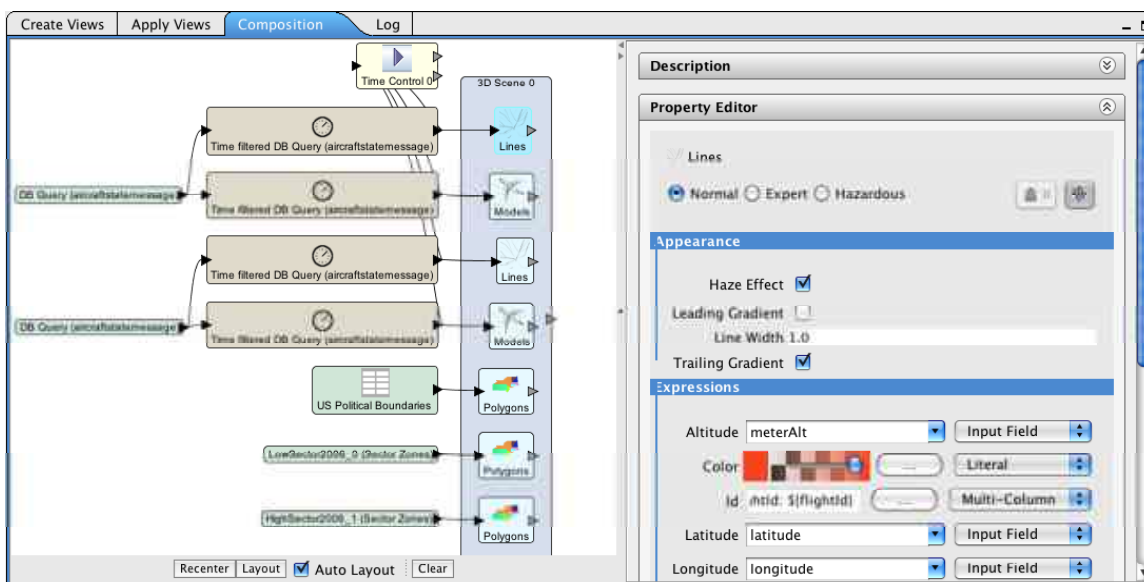


Figure 4. NAS Viewer Configuration User Interface. To specify a visualization, a user drags entities from the right side of the screen, such as a Renderer, into the Composition Graph[§] on the left, and then uses other entities on the right, such as Expressions, to specify properties, such as color. Users must also properly add in links (arrows) between entities on the left.

III. Interactive Visualization of NAS Data in 4 Dimensions — IV4D

The goal of the IV4D development team was to produce a data visualization tool that provides researchers with an effective way to study and understand activity in the NAS using a wide variety of forms of input and output data. Ideally it would be easy for the researchers to understand and use with little training, and easily adapted to meet new data and visualization needs. The NASA development team concentrated on tailoring IV4D for the NASA aeronautical research community by defining and implementing an interface between the researchers and the 3D Scene display of the NAS Viewer (Fig. 2). NASA developers also worked with AFRL to specify NAS Viewer enhancements. These included handling data more generically, loading and

[§] A ‘scene graph’ is a general data structure used by programmers for vector-based graphics. Because the NAS Viewer’s internal scene graph is represented on the Composition panel (i.e., it is a diagram of the current internal program structure) it is referred to as a Composition Graph.

running large data sets with more than 100,000 flights, and modifying transparency and translucency in visualizations that show thousands of objects at a time.

As shown in the notional architecture design diagram in Fig. 2, IV4D extends, but does not replace, the capabilities developed for the JView and NAS Viewer tools. To address the IV4D design requirements, the NASA developers added the Create View and Apply View software layers. The term “View,” is analogous to a template for creating visualizations. It refers to meta-data saved and used by IV4D to create and recreate a particular type of visualization. This meta-data can be applied to any appropriate data input to IV4D. For example, a user can create and save the meta-data filters used to color flights departing from Phoenix red from the “Create View” panel. Once saved, users can then apply the meta-data from the “Apply View” panel to any input data containing flights departing from Phoenix, and these flights will be colored red. Existing functionality is retained and the user can still access the Composition panel (Fig. 4), which requires expert knowledge of data visualization techniques, or rote memorization of numerous steps to produce a result such as that shown in Fig. 3. However, with the Create and Apply View functionalities, manipulation of the Composition panel is rarely needed or requires many fewer steps.

A. Create View

The Create View functionality expands on the existing data parsing and selection capabilities built into the NAS Viewer by addressing the first three IV4D design requirements:

1. Determine data formats to visualize
2. Create an easy to learn user interface that finds data to visualize
3. Develop heuristics for determining appropriate visualization components, based on user selection

1. Data Formats

Since researchers were using IV4D while it was being developed, it was important to incorporate data that was immediately available, in particular ACES data, in its current format rather than attempt to define a more generic data format. Though IV4D inherited a data interface from the NAS Viewer, it enhanced the software to actively look for and parse data from sources of interest to ATM researchers, especially researchers working with ACES. By having IV4D search for and focus on the simplest set of data needed for creating visualizations of NAS data in JView, IV4D provided an easy way for researchers to view arbitrary NAS scenes.

The minimal dataset necessary for IV4D to visualize 4D ATM data initially came from the ACES simulation. ACES data is saved as comma separated value (CSV) data files or relational database files. The CSV data contains airspace boundary data, which is assumed to represent static airspace volumes unless there is a data field that represents time. Airspace boundary data may also represent areas of weather, special use airspace such as military no-fly zones, or other 3D volumes. These data typically include a name or other type of identification, floor and ceiling heights, and a series of latitude/longitude points that describe the area of interest. ACES relational database files contain output from analysis runs with aircraft locations and trajectory predictions, typically latitude/longitude/altitude positions. Relational databases such as MySQL or Oracle can support complicated data structures— using column(s) of key values in one table that point to data in other tables, for example. The most basic data format in a relational database is a simple table containing rows and columns. An advantage in using a relational database over a file containing, for example, comma delimited plain text, is that structure query language

(SQL) queries can be used to extract and manipulate segments of relational data into a new table, called a result set, which contains only the data needed for a particular purpose.

The 4D aspect of the data format is introduced by using time associated with position data. Flights and weather cells are commonly tracked through their latitude, longitude, and altitude positions at regular time intervals. To show the movement of these attributes with time, IV4D requires a consistent name or identification for the item and the inclusion of a time field in the data. The software accepts time in units greater than one millisecond, though time is typically generated in seconds or minutes by the researchers. Using the underlying JView technologies, data points representing a specific time can be displayed in succession, thus animating the data.

Many types of flight data are commonly displayed as line segments. A route depicts the 2D horizontal path an aircraft is expected to fly; while a trajectory combines the horizontal path with altitude changes over time representing a 4D prediction of an aircraft's path. Observed position reports can also be connected to provide a history of an aircraft's flight. Though each can be represented by IV4D as a series of line segments, their meaning and how they are presented to the user impacts the value of a visualization. In addition, ATM software will often generate multiple trajectories for an aircraft based on a given position in order, for example, to advise an air traffic controller of a flight plan amendment. Hence, it is possible to have many display elements for a single flight – the originally filed route, the observed position reports, and a set of trajectories generated by ATM algorithms during flight. In this case, the user may want to display the moving flight position over a static trajectory to understand the difference between the predicted and observed flight path. Fig. 5 shows a visualization of an aircraft with its pre-flight trajectory as well as its proposed flight plan amendment trajectories generated to keep the aircraft separated by a predefined distance and altitude, represented by the orange and magenta cylinders.

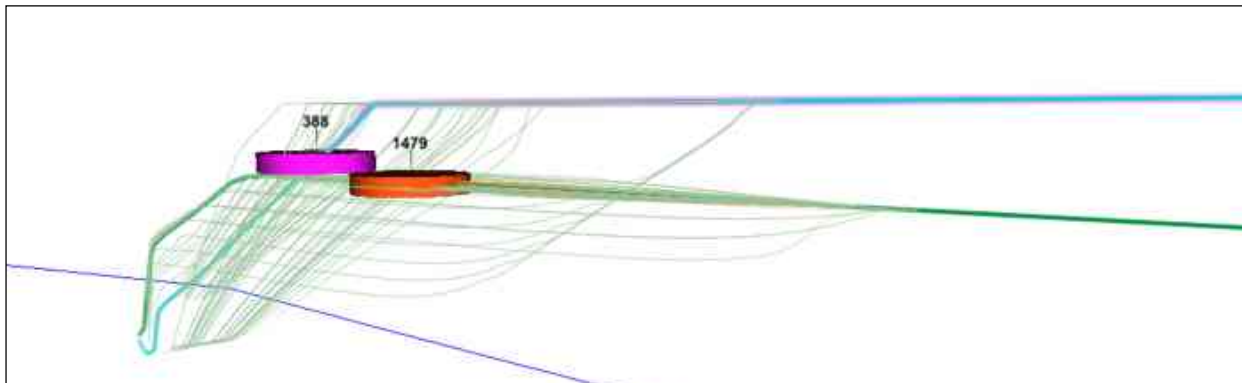


Figure 5. Trajectory Visualization. *Flights are represented as cylindrical zones that must not intersect (separation zones). Flights are moving from the right to the left. Alternate trajectories, called trial plans, were calculated when the prediction algorithm warned of a possible loss of separation. In this example, the loss of separation was avoided through a speed maneuver rather than through a route maneuver.*

Once a functional version of IV4D using the minimal dataset of ATM data was available, IV4D was expanded to visualize more complex data and data from external sources. NASA and AFRL developers modified the NAS Viewer API to handle data not in the minimal data set, for example whether an aircraft is conforming to its intended route of flight. Working with external data sources, NASA developers and IV4D users found it easier to write simple translation scripts

to convert these data into the existing format expected by IV4D. For instance, the FAA’s Aircraft Situation Display to Industry (ASDI) data contains the same, or similar, information as ACES data including unique flight IDs, timestamps at regular intervals, latitude and longitude, and elevation.¹² Simple scripts that translate ASDI data and CTAS simulation data, as well as scripts that display output from NASA Langley Airspace and Traffic Operations Simulation (ATOS) data runs have been written¹³ and used to visualize the data. The use of scripts allows analysts who are familiar with the data to perform the data translation. (An example of visualizing translated data is shown in Fig. 21 in the Appendix.)

2. Create Views User Interface and Data Search Heuristics

Visualization is an important tool for data mining, but before the data can be displayed, it must be selected from the available information. To that end, IV4D provides a user interface that presents data selection and filtering mechanisms modeled on standard user interface solutions, such as menu selections, text boxes, etc. The amount of data that can be displayed from data sets can be overwhelming. IV4D provides tools to display only those aspects of the data that are important. The goal is to give users a way to create useful visualizations quickly.

Once the data set has been selected, if it is one of the well-known data sets such as track data from an ACES analysis run, it can be immediately read into the program and displayed on the JView display canvas. If it is a new data set, or has more data than a strictly minimal set, IV4D provides a mechanism to select appropriate data elements. Figure 6 shows an IV4D panel that allows the user to select from among several data containing values for latitude. Latitude is a mandatory value, but when several ‘latitude’ columns are present in the data, the user can use IV4D’s selection or pick the correct value to include for visualizing.

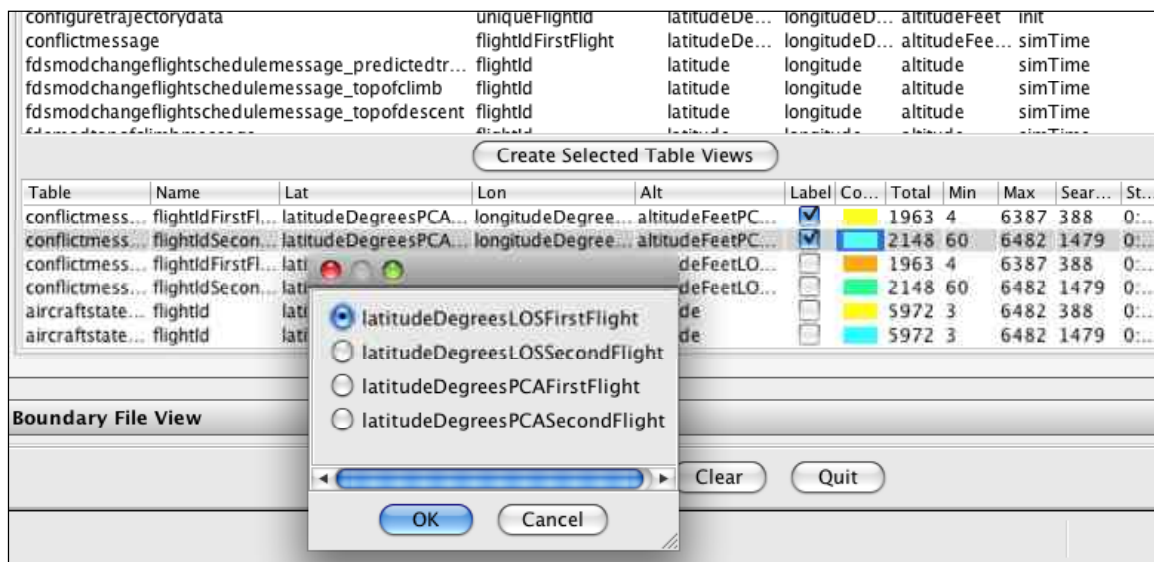


Figure 6. IV4D select desired latitude column. When there is a choice, users can specify the desired value. Besides latitude, users can select correct values for ID, longitude, and altitude.

Furthermore, IV4D provides heuristics to associate visualization data elements with actual data. For instance, when identifying trajectory data, IV4D looks for the most likely “identification” candidate data schema name in the following heuristic order:

- 1) flightId
- 2) Anything containing “flightId,” such as “uniqueFlightId,” “etmsFlightId,” or “airlineFlightId” (pick the first found, if there is more than one)
- 3) etmsId (from the FAA – enhanced traffic management system Id)
- 4) airlineFlightNumber
- 5) The first column name that ends in “id,” such as “AIRPORT_ID” or “airportId”
- 6) The first column name that ends in “name,” such as “fixedWingName”

Since the trajectory data comes from a relational database, the database contains one or more distinct sets of data that, like CSV data, can be displayed in columns and rows and are therefore referred to as tables. A resulting benefit is that users looking at large or new data sources, especially for the first time, need not know in advance which tables will produce a visualization: they can explore the database and IV4D will list the available tables. The user then selects one or more of the listed tables to produce a visualization.

Once tables are selected, users can apply filters and attributes to the data that determine what elements are seen and how they are depicted in the visualization. Example filters include selecting flights arriving to a particular airport or selecting regions of airspace, such as a particular set air traffic controlled regions, referred to as ‘sectors.’ Attributes include color, whether or not to show ID text labels, and start or stop times. IV4D provides three mechanisms for selecting objects for filtering:

- 1) Data denoted by values separated by commas (,), and/or ranges from a – b;
- 2) Regular expressions; and
- 3) SQL database queries - it is also possible to add data from other columns or tables by using a SQL query.

Once all of the data and filtering selections have been made, IV4D allows the user to save this data view. The next time this data set is selected, the same view can be reapplied and the data will be visualized automatically. Figure 7 shows the resulting visualization after a SQL database query to select a set of flights arriving and departing Phoenix.

3. Data Visualization and Visualization Heuristics

Developing a visualization tool is a balance between ease of use and the capability to produce satisfactory displays. The JView graphics engine allows nearly unlimited visual configurations and the NAS Viewer allows users to set hundreds of display options, while IV4D presents users with a more constrained set of choices. Nonetheless, all three software layers (Fig. 2) lead to visualizations that can be played back. To help users of IV4D obtain results quickly, IV4D allows objects to be filtered for visualization (by name, using regular expressions, using SQL queries, by time interval, etc.) but limits the display options to a standard set often used for ATM research (for simplicity).

This feature presented a challenge for the IV4D developers since a “good” visual display often requires a number of refinements that make it pleasing or easy to view. Whereas most graphics applications add settings to pass this responsibility on to the user, IV4D employs a set of heuristics that make assumptions about what the user wants to see, based on the data the user selects.

Fig. 7 shows the default visualization for a basic set of track data arriving to and departing from Phoenix. Because IV4D assumes researchers are interested in NAS-wide flight behavior

(i.e., airspace volumes and flights over the continental U.S.) IV4D always includes a map showing continental U.S. state boundaries. In addition, IV4D:

- Exaggerates altitude by a scale of 10:1
- Displays animated flight objects using a model that resembles a paper airplane
- Adds a 5-minute history trail behind each animated flight object
- Specifies label dimensions, placement with respect to its object, and font, when present
- Uses a graduated pin to connect the label to a flight or to a waypoint
- May draw complete trajectories as solid lines that connect waypoints along a trajectory's path. (This depends upon a combination of id and timestamps across rows of data.)
- Sets the default playback time-step to 1 minute

There is more, but in essence the idea is to assume that the best view represents a broad geographical area.

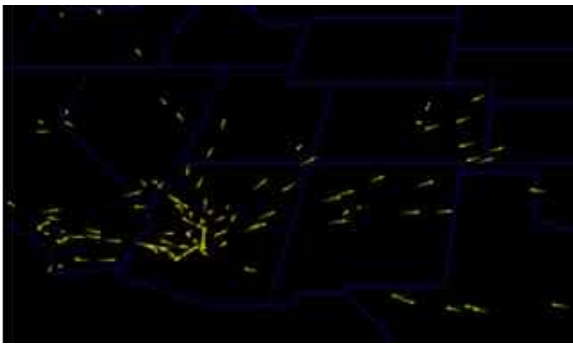


Figure 7. Default view of flight track data.
Taken from a simulation of a day's worth of traffic. Only flight tracks for arrivals to and departures from Phoenix are visualized.

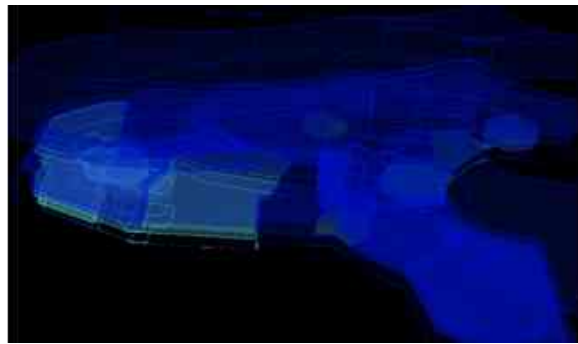


Figure 8. Default view of airspace data.
Taken from CSV data that models Low and High sectors for Albuquerque Center in 2006.

Fig. 8 shows the visualization for a set of typical airspace regions, called sectors, and includes the default U.S. boundary map. For a given airspace region, floor and ceiling polygons are outlined, so the viewer sees a plane for the floor and another for the top. The ceiling plane is filled with a translucent color, while the floor and the sides, known as “shelves,” are transparent. The effect looks somewhat like stained glass, especially when there are many layers, as is often the case. (Additional examples of tracks and airspace can be seen in Figs. 13 – 20 in the Appendix.)

B. Apply View

As a scene grows in complexity, so do the heuristics to visualize the scene. For example, Fig. 9 shows a pair of aircraft together with their respective originally predicted (pre-flight) trajectories. The aircraft positions and color came from one set of data saved as a view, while the trajectory came from a separate view. In this case, the aircraft track history color, yellow seen over red for ID 388 and blue seen over magenta, for ID 1479 (Figure 9), was modified so it would show over the pre-flight route. The IV4D Apply View feature allows a visualization to be built as shown in Fig. 9. This functionality enhances the usability of the tool and addresses the final IV4D design requirement (others were addressed previously): allow a user to quickly apply the settings from an existing visualization to new data, and allow a user to mix together and modify different visualizations.

1. View Modification

Default visualizations for ATM data provided by IV4D provide basic views, which accurately display selected objects. When creating scenes for display, users may need to refine or improve a visualization. IV4D constitutes a layer of software that depends upon and uses an API from the NAS Viewer, which, in turn, depends upon and uses an API from the AFRL JView graphics engine (Fig. 2). For IV4D to successfully fulfill its purpose, AFRL reengineered parts of the NAS Viewer to allow default filters and attributes used by IV4D for visualization to be more readily modified.

Using the NAS Viewer's Composition panel (Fig. 4) to build a display configuration involves selecting and specifying each data set, filter, and attributes the user needs for the final visualization. As mentioned previously, IV4D selects default values for a wide array of ATM-related data and a Composition Graph is produced automatically. Characteristics of a visualization may be changed by simply modifying an existing Composition Graph. Figure 4 shows the Composition Graph produced when creating the visualization shown in Fig. 10. To make further changes to the visualization, the user simply highlights a part of the graph and changes property settings using the Property Editor on the right (Fig. 4). (Additional examples of using the Property Editor can be seen in Figs. 17 – 20 in the Appendix.)

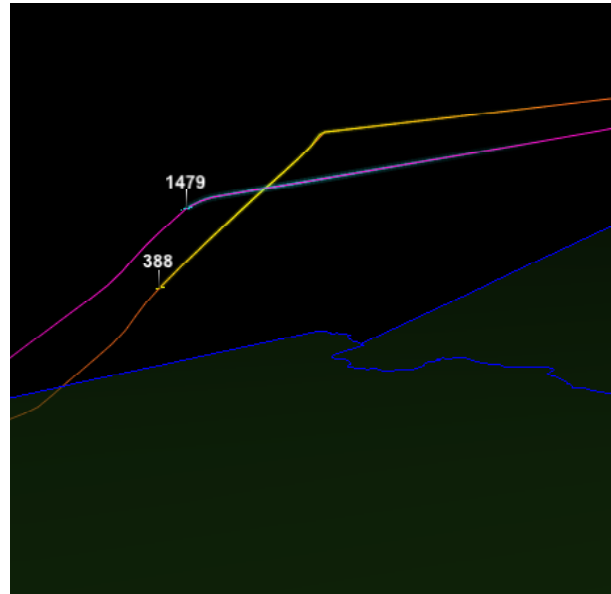


Figure 9. Default Visualization of Trajectory Data. This view shows a pair of flights, along with their original, pre-flight, route trajectories.

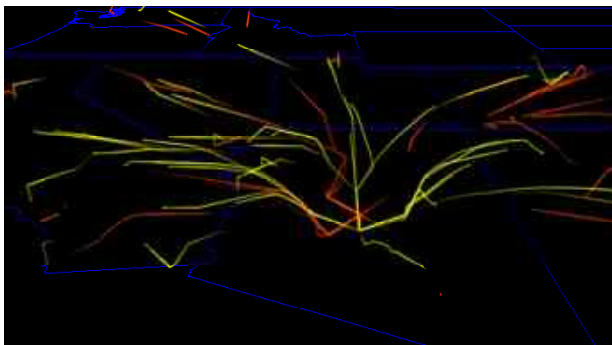


Figure 10. Modified Track Visualization from IV4D. A modified visualization of Albuquerque Center air traffic data. Arrivals to Phoenix are in red.

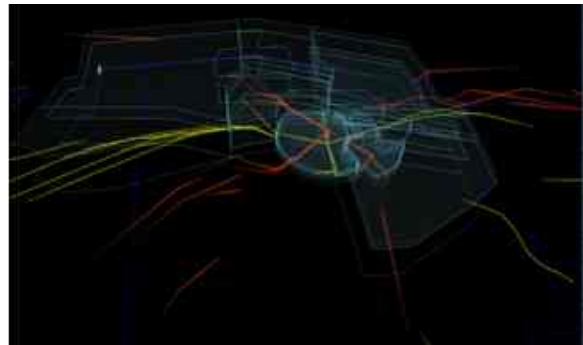


Figure 11. Combined Track and Airspace Visualization from IV4D. A modified visualization of Albuquerque Center air traffic data. Aircraft and sectors are shown.

The underlying NAS Viewer code also allows the user to employ even more advanced data filtering via simple scripts that can be accessed through the Property Editor. “Extra” data not

automatically parsed by the IV4D parsing logic can be assigned values and displayed dynamically using the Property Editor. This is the core NAS Viewer capability that IV4D was designed to hide from the casual user, but it is, nevertheless, still available. For example, Fig. 12 shows the same Albuquerque Center aircraft as seen in Figs. 7, 10 and 11, but with additional color and longer trails: arrivals to Phoenix are in cyan and departures from Phoenix in yellow. Flight trails are lengthened to show 2 hours of flight, and whenever a flight is maneuvered off route, cyan becomes red, while yellow changes to magenta. This is an updated visualization from Fig. 7, with scripted filters applied via the Composition panel. (Additional examples of using scripting can be seen in Figs. 17 and 20 in the Appendix.)

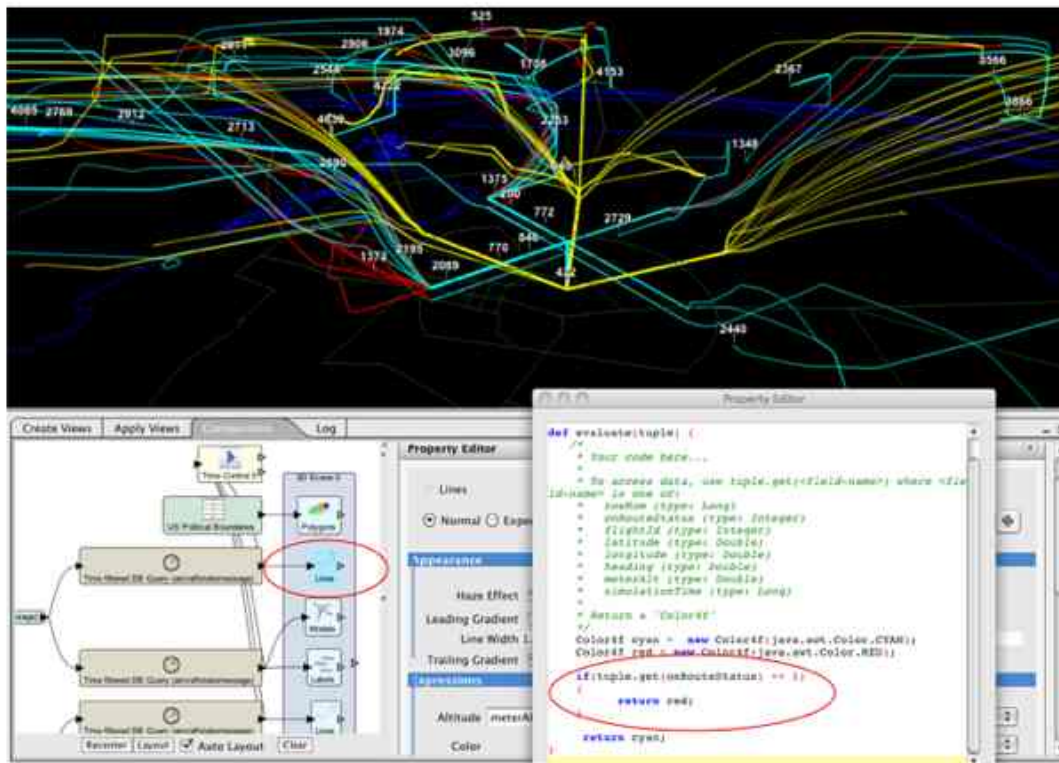


Figure 12. Scripting track color. Flights arriving to Phoenix are scripted to turn red when they are maneuvered off-route in order to maintain separation from other flights. A two hour trail is kept on screen, to better show rerouting.

2. Mix and Match Visualization and Meta-Data (Applied Views)

With the ability to quickly generate default visualizations for ATM data and to modify the views with the underlying NAS Viewer capabilities, the true benefit of the IV4D is in applying views to new data sets and combining views to generate new visualizations. Besides saving visualization settings, IV4D allows users who have saved settings for a particular visualization created from the Create Views module to use the Applied Views module to:

- 1) Replay a saved visualization;
- 2) Apply a saved visualization settings to new data;
- 3) Replay multiple visualizations together (visualizations that were not created together); and

4) Apply settings from combined visualizations to new data.

For example, the saved visualization settings for the airspace sectors from Fig. 8 were combined with the filtered set of Phoenix arrivals from Fig. 10 to create the visualization shown in Fig. 11. Also, in this case, the arrival flights data are selected from a different day and the airspace data was filtered to show only low-altitude sectors. Once the original views are created and saved via the Create View module, the combined view may be developed easily.

IV. Conclusion

Designed for use by aeronautical researchers and consumers of aeronautical data, and available for general use, IV4D's main goal is to help users set up and quickly see accurate, esthetically pleasing, default visualizations from ATM data. Researchers and analysts, including non-programmers and student interns, at NASA's Ames and Langley Research Centers use IV4D to visualize ATM data. Little or no programming knowledge is needed and, because IV4D is designed to read a simple set of table data, in columns and rows, the visualizing application remains independent from particular source data. IV4D has been used to visualize data from ACES, (archived) track and flight plan data from the CTAS software application, human-in-the-loop data from piloted separation assurance trials at NASA Langley, and ASDI and airspace boundary data from the FAA. Users are able to simply extend their default visualizations by utilizing AFRL's NAS Viewer and JView advanced visualization technologies.

IV4D's secondary goal is to give users the capability to apply existing settings to new data and to combine visualizations that were created separately. These features allow the user to create and observe "what if" scenarios or visually compare and contrast scenarios without setting up new simulations or spending time writing new software.

Future enhancements to IV4D might include improving IV4D's display heuristics. For example, when IV4D determines that objects are dispersed broadly or narrowly, or that the data is "crowded" with many overlapping objects, the default visualization could be improved by zooming in or out, or by adjusting translucency of the displayed objects. Another improvement might be to have IV4D manage CSV data more flexibly as it does the SQL data.

Appendix

Many of the figures used throughout the paper were simplified in an attempt to illustrate the functionality of IV4D without overwhelming the reader with overly complex graphics. Following are a series of figures generated by the IV4D tool combining typical views with simple filtering.

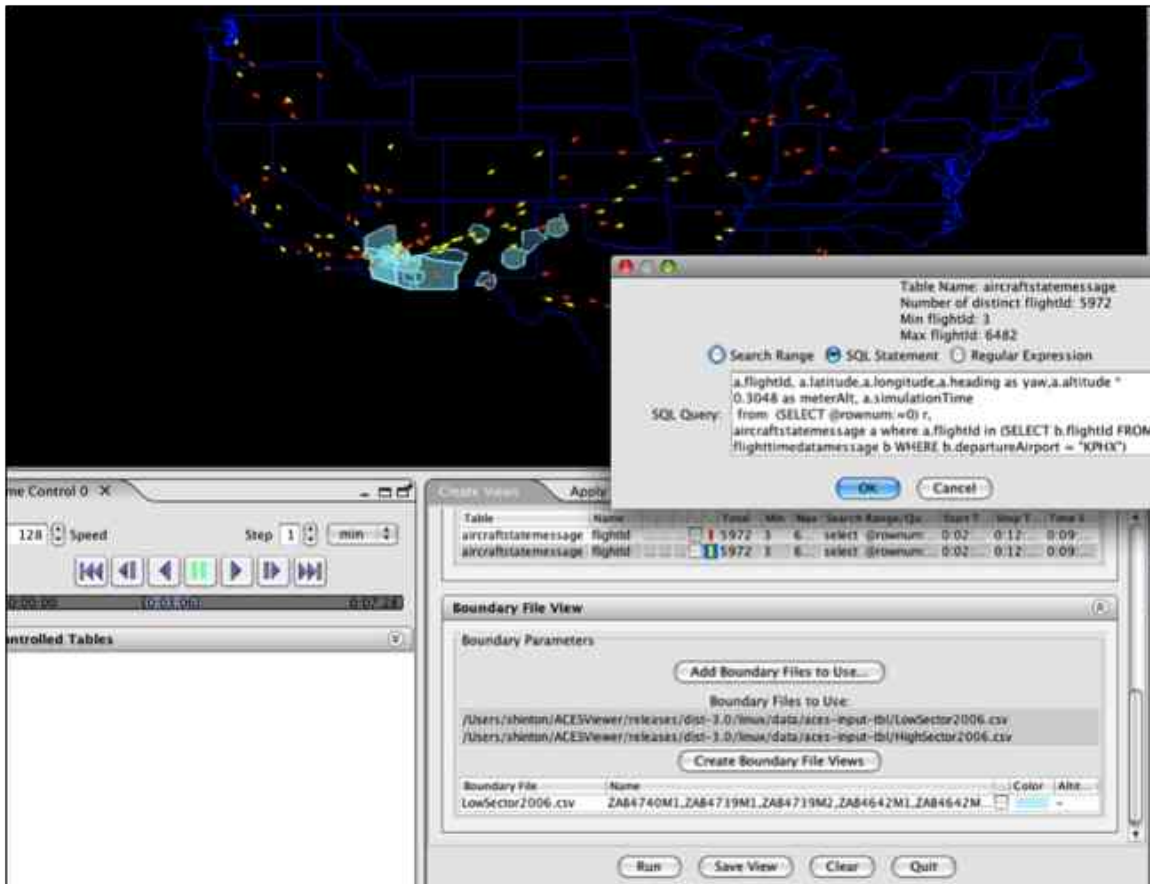


Figure 13. Filtered data. SQL flight data showing flights to (red) and from (yellow) Phoenix. CSV airspace boundary data showing only 2006 Low Sectors for the Albuquerque Center.

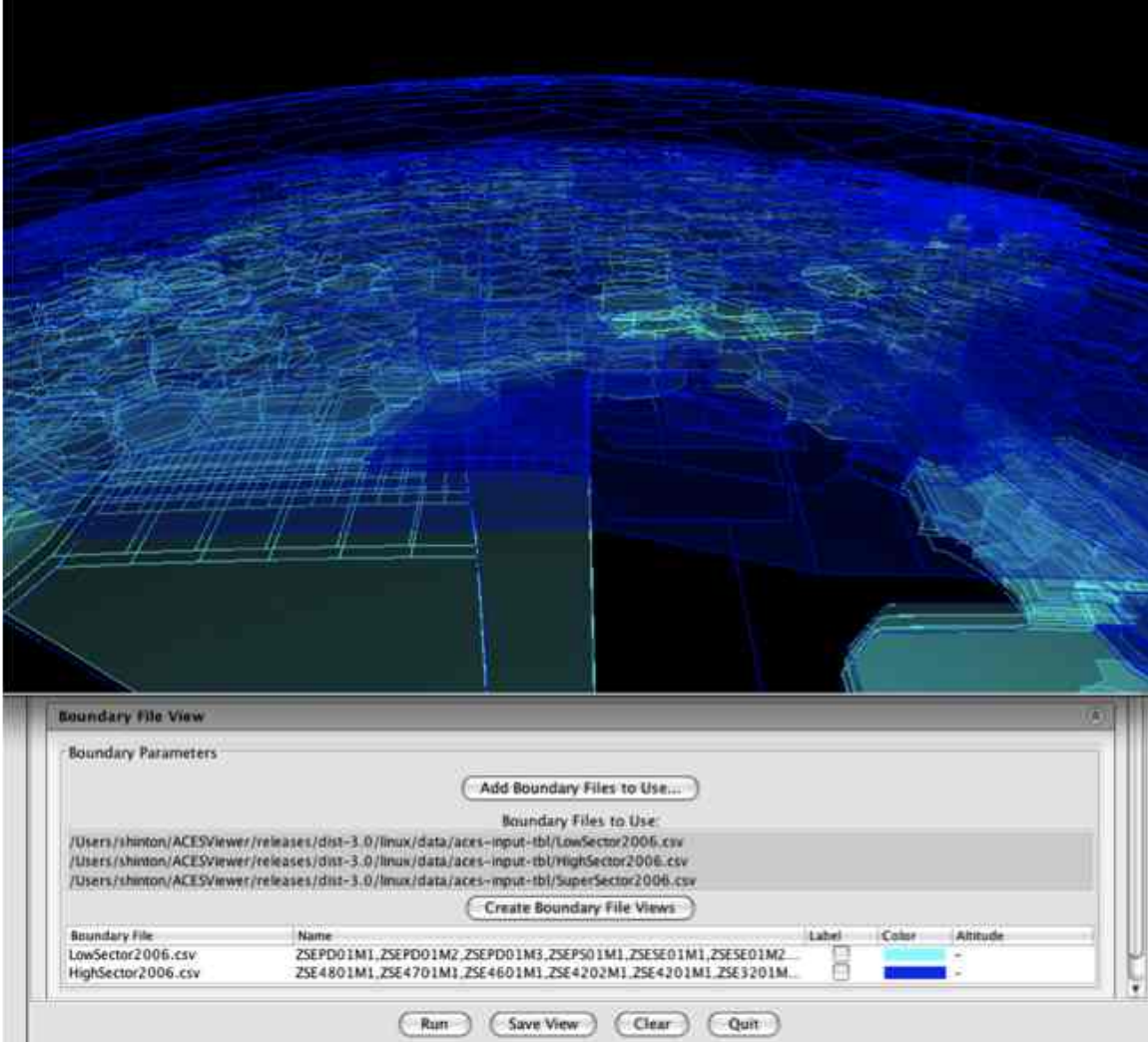


Figure 14. Default view of airspace data (WB584 Projection selected). Taken from CSV data that models 2006 Low and High sectors used by the FAA.

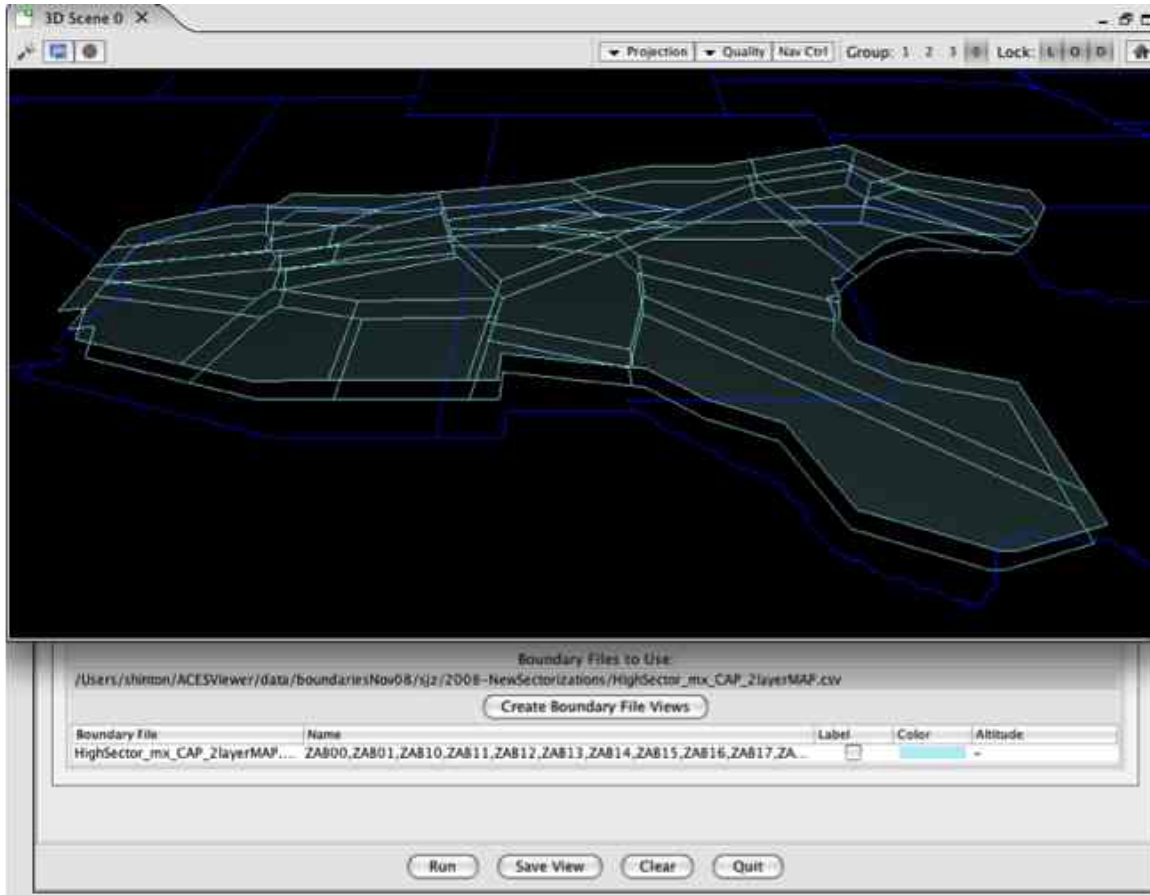


Figure 15. Mathematically created sectors. Taken from researcher created data that models High sectors by running flight tracks through a Voronoi inspired algorithm. These sectors are not used in the real NAS.

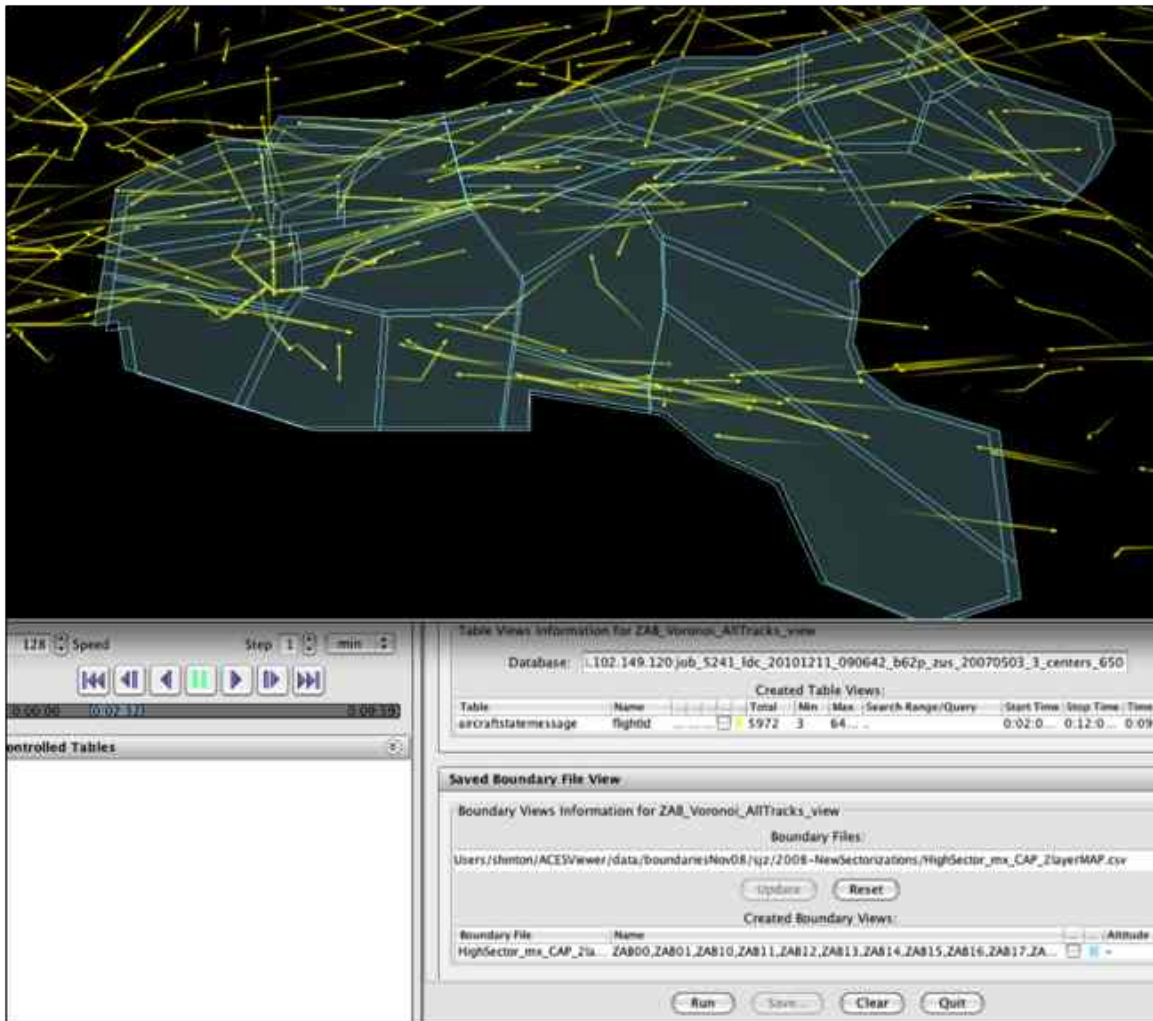


Figure 16. Apply Views combining Voronoi airspace and earlier flight tracks. *Though the flight track data came from one study and the Voronoi airspace came from another, here they are combined. It is easy to see that though some of the tracks fit well with the made up airspace, other tracks do not. Apply Views makes it easy to see many sets of flight tracks with the new airspace, without needing to run new simulations.*

(Classical Voronoi diagrams divide space in order to define a set of points closest to a particular point, especially with respect to other points – i.e., into ‘cells’ where cell boundaries are equidistant from a specified point within each cell. When dividing airspace for air traffic, the cell boundaries are adjusted for broader, dense, areas representing flight tracks over a given time period, such as over 1 day.)

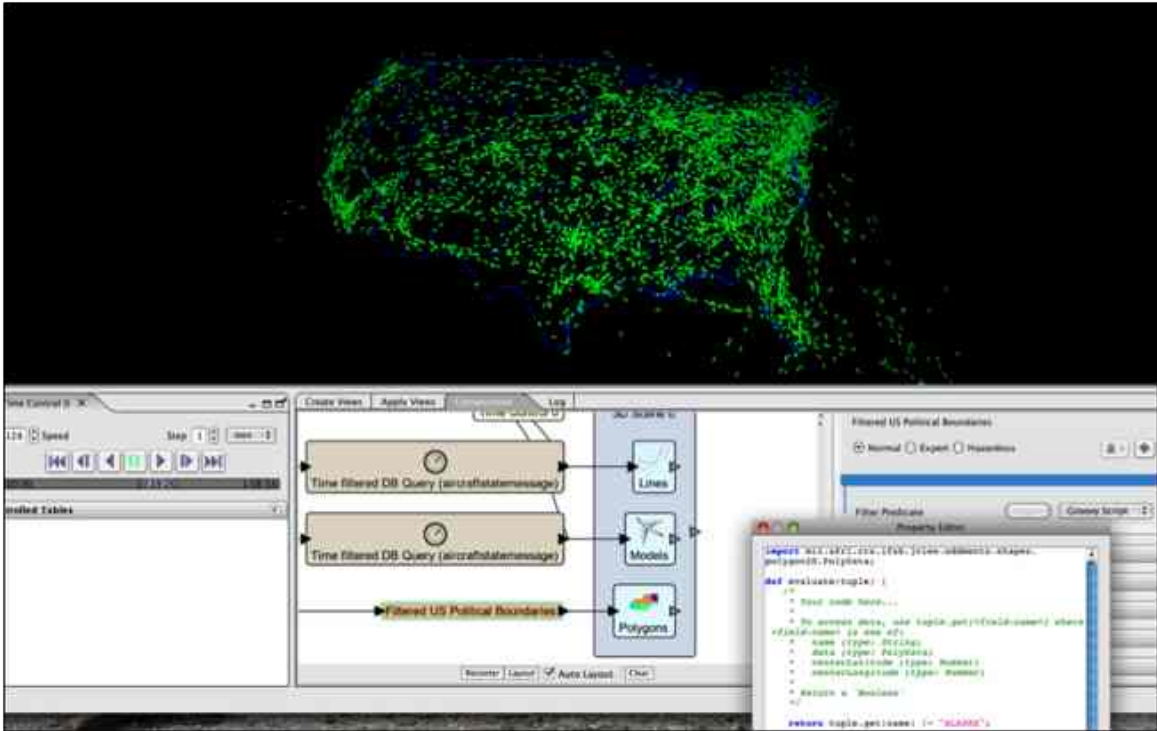


Figure 17. Using a Groovy script to remove the state of Alaska. After adding a filter to the Composition Graph, for U.S. Political Boundaries, a user can add a code snippet to change the display. In this case, the boundary data for “ALASKA” is filtered out.

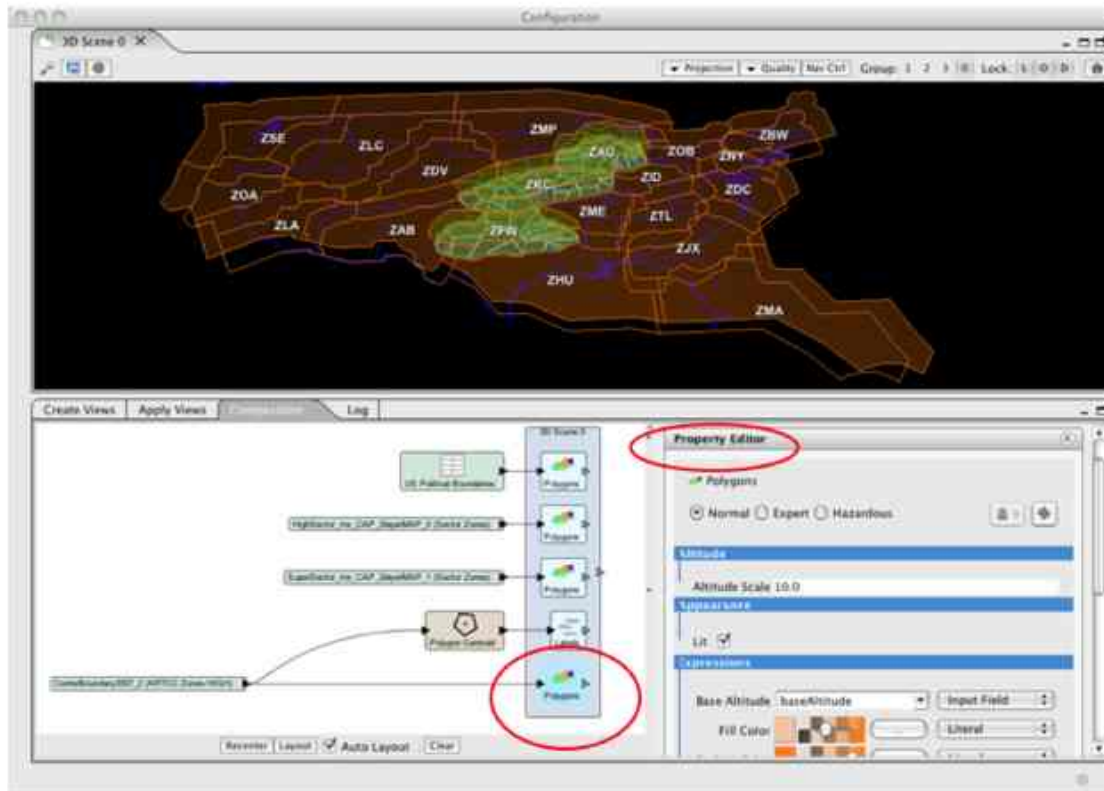


Figure 18. IV4D produces a NAS Viewer Composition graph along with a visualization. The user's goal was to show how new Voronoi airspace fits into the existing Center boundaries. Though the visualization is reasonably clear, it can be made better by changing a few properties.

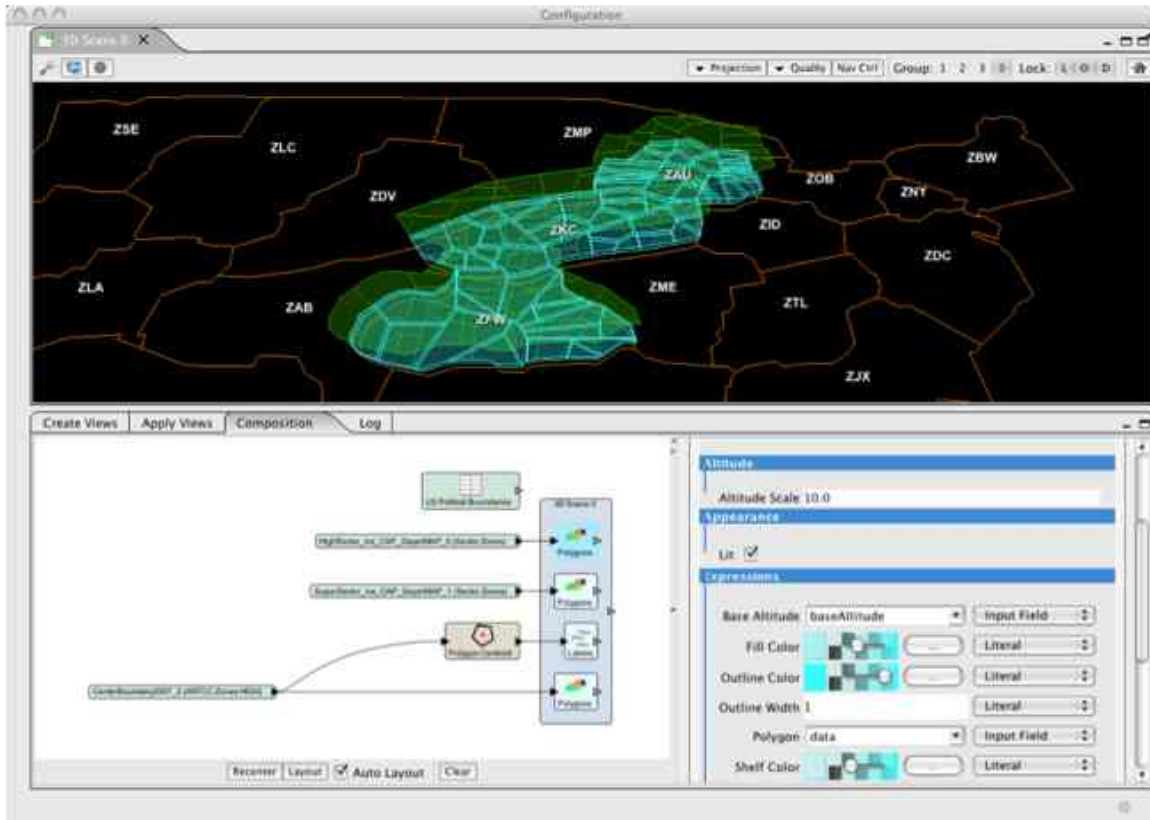


Figure 19. Editing the NAS Viewer property settings to change the visualization. Flattening out the orange Center boundaries and changing colors in the Voronoi airspace areas makes this picture easier to view and understand.

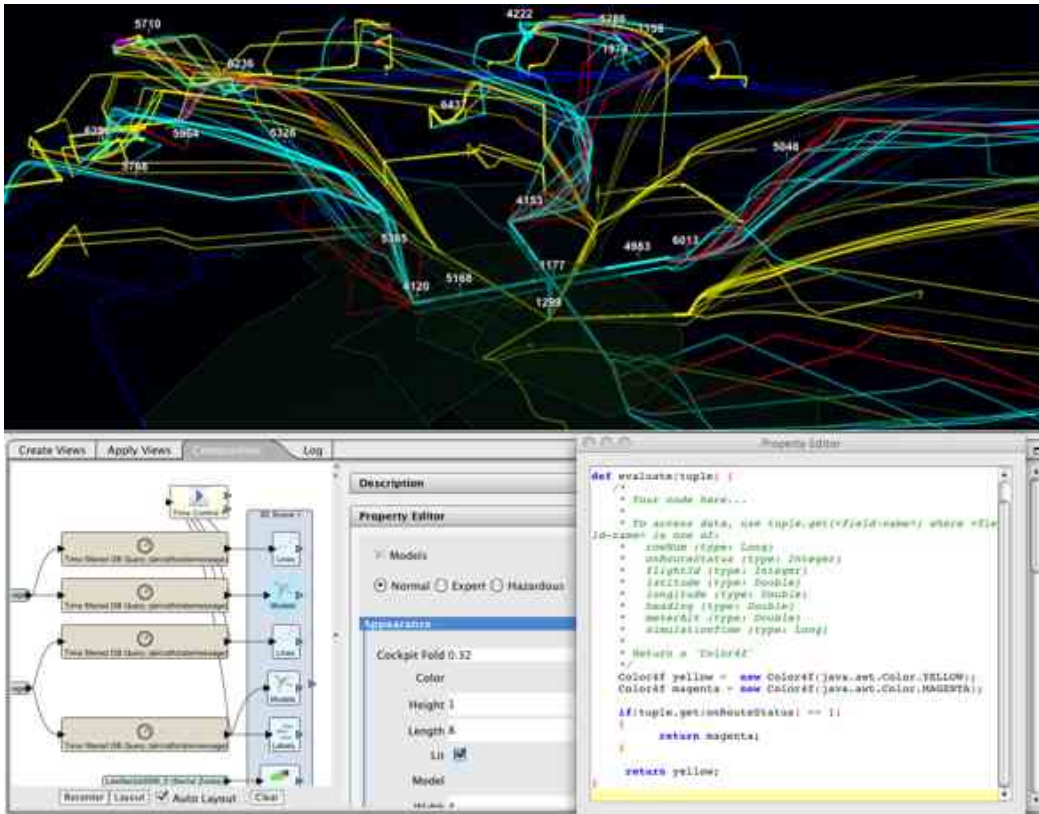


Figure 20. Scripting track color. Flights arriving to Phoenix are scripted to turn red when they are maneuvered off-route in order to maintain separation from other flights. Departing flights maneuvered off-route are scripted to turn magenta. A two hour trail is kept on screen, to better show rerouting.



Figure 21. Dallas-Fort Worth Runways. *FAA Runway data was translated into a format normally used by IV4D for visualizing a complete trajectory as a line. In this case, altitude was set at 0 feet, while latitude and longitude “waypoints” formed straight lines.*

Acknowledgements

The authors would like to acknowledge software developers Daniel Krisher, who worked at AFRL on NAS Viewer software and Xin Cao and Dmitry Faradjev, who worked at NASA Ames Research Center on IV4D.

References

-
- ¹ Green, S. M., and Vivona, R. A., "En route Descent Advisor Concept for Arrival Metering," AIAA-2001-4114, Guidance, Navigation, and Control Conference, Montreal, Canada, Aug. 2001.
 - ² Swenson, H. N., Hoang, T., Engelland, S., Vincent, D., Sanders, T., Sanford, B., and Heere, K., "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France, June 1997.
 - ³ Davis, T. J., Krzeczowski, K. J., and Bergh, C., "The Final Approach Spacing Tool," IFAC Thirteenth Symposium on Automatic Control in Aerospace, Palo Alto, CA, Sep. 1994.
 - ⁴ Coppenbarger, R. A., Lanier, R., Sweet, D., and Dorsky, S., "Design and Development of the En Route Descent Advisor (EDA) for Conflict-Free Arrival Metering," AIAA-2004-4875, AIAA Guidance, Navigation, and Control Conference, Providence, RI, 16-19 Aug. 2004
 - ⁵ Sridhar, B., and Sheth, K., "Visualization of Air Traffic Flow for Modeling and Control Applications," International Federation of Automatic Control (IFAC) 17th World Congress, Seoul, Korea, 6-11 July 2008
 - ⁶ Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool" 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, Italy, 13-16 June 2000.
 - ⁷ Sweet, D., Manikonda, V., Aronson, J., Roth, K., and Blake, M., "Fast-Time Simulation System for Analysis of Advanced Air Transportation Concepts," AIAA-2002-4593, AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, CA, Aug. 2002.

⁸ Meyn, L., Romer, T., Roth, K., Bjarke, L., and Hinton, S., "Preliminary Assessment of Future Operational Concepts Using the Airspace Concept Evaluation System," AIAA-2004-6508, AIAA Aviation Technology, Integration, and Operations (ATIO) Forum, Chicago, IL, Sep. 2004.

⁹ "Advanced Visualization and Interactive Displays (AVID)", AFRL-RI-RS-TR-2009-115, Air Force Research Laboratory, April 2009.

¹⁰ Hinton, Susan, "Final Technical Report Interactive Visualization of NAS Data in 4D (IV4D)" FA8750-07-0050, Air Force Research Laboratory, May 2010.

¹¹ <http://earth.google.com>

¹² "Aircraft Situation Display To Industry: Functional Description and Interface Control Document for the XML Version", Version 1.7, U. S. Federal Aviation Administration, May 2009.

¹³ Peters, M., Ballin M., Sakosky J., "A Multi-Operator Simulation for Investigation of Distributed Air Traffic Management Concepts", AIAA-2002-4596, AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, CA, Aug. 5-8, 2002.