AFRL-IF-RS-TR-2006-6
Final Technical Report
January 2006

VISUALIZATION FOR INSIGHT INTO THE OVERALL NAS (VISION)

HRL Laboratories, LLC

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2006-6 has been reviewed and is approved for publication

APPROVED: /s/

TYRONE J. BALMACEDA, 2Lt, USAF
Project Engineer

FOR THE DIRECTOR: /s/

JAMES W. CUSACK
Chief, Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JANUARY 2006	Final Sep 2004– Sep 2005

4. TITLE AND SUBTITLE
VISUALIZATION FOR INSIGHT INTO THE OVERALL NAS (VISION)

5. FUNDING NUMBERS
C - FA8750-04-C-0277
PE - NA
PR - NASA
TA - BA
WU - 03

6. AUTHOR(S)

Dr Ronald Azuma

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

HRL Laboratories, LLC
3011 Malibu Canyon Road
Malibu California 90265

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFSB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2006-6

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Tyrone Balmaceda, 2Lt.,USAF/IFSB/(315) 330-2115/Tyrone.Balmaceda@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

The VisION project presents 12 new visualization concepts for generating insight and understanding from National Airspace System simulation data, along with three preliminary and initial evaluations for those. Researchers developing new approaches to increase NAS capacity need improved visualization techniques to better understand the simulated results of their concepts. We instead develop novel visualization approaches that do not rely primarily upon plotting data at their true geographic coordinates. The goal is to generate insight and allow observers to find important characteristics that they did not even know to look for initially. The report documents our approach and strategies, ideas considered but abandoned, and the 12 concepts that reveal different aspects of NAS simulation data. A separate software description report documents the three implemented concepts. A separate evaluation report documents the observations and lessons learned. We documented cases where the visualization modes enable us to see patterns, trends, correlations, features, and relationships that we were previously not aware of.

14. SUBJECT TERMS
Information visualization, air traffic management, congestion, delay, correlations, simulation, capacity enhancement.

15. NUMBER OF PAGES
49

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

List of Appendixes

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

This is the final report for the VisION project. This document satisfies the contract deliverable requiring a final project report.

VisION was a year-long, $300K project funded by AFRL Rome to develop novel visualization ideas for understanding the output from simulations of the National Airspace System. The primary output is the set of generated ideas, where the ideas do not rely primarily upon accurate geographic plotting of information. The secondary outputs are the simulation datasets, preliminary evaluations of some of the concepts, and the software implementing the basic characteristics of some of the visualization concepts. HRL Laboratories, LLC was the prime contractor, with Raytheon as a subcontractor to provide the simulation datasets.

The VisION contract required a minimum of three developed concepts and one preliminary implementation. We provide much more than this minimum.

The first part of this final report gives a brief overview of the project: the motivating factors, goals, and approaches and strategies used to develop the novel visualization concepts. Then the middle of the report describes the heart of the project: twelve visualization concepts, each focused on illuminating different aspects of the NAS. Then we briefly describe the ACES simulation datasets that Raytheon provided, and summarize the evaluation and software implementations. Both evaluation and software implementations are covered in their own separate reports (VisION evaluation report and VisION software description report) so these are only summarized here. Finally, we offer conclusions and describe future work that can be done in this area.

We published and presented a paper at the AIAA 2005 Modeling and Simulation Conference (in San Francisco, August 15–18, 2005) entitled “Visualization Concepts for Generating Insight from NAS Simulation Data.” Leighton Quon, the NASA employee who is second in command of the VAMS project, chaired that paper session and indicated that he liked our concepts and was interested in seeing them further developed and made available to the VAMS concept developers.

The major team members performing on the VisION contract were Ronald Azuma, Mike Daily, Tim Clausner and Jason Fox of HRL Laboratories, and Mary Ellen Miller and Greg Trott of Raytheon Company.

Overall, we successfully completed all of the contract requirements. In the evaluation, we were able to provide evidence of the types of features that we expected each visualization to make apparent. We also found examples of previously unknown characteristics and items that we did not know to look for. We managed to do this despite the limited number of ACES datasets available to us. However, this work really needs a follow-on effort to build interactive versions of these visualizations, integrated with other data visualization tools, that end users can use to visualize the results of simulations of their own capacity-enhancement concepts.

2.0 PROJECT DESCRIPTION AND GOALS

2.1 Motivation and Rationale

The amount of traffic in the National Airspace System (NAS) is expected to double by the year 2025. Meeting this projected growth in demand requires innovative new proposals for future Air Traffic Management (ATM) systems. Because of this need, NASA initiated the Virtual Airspace Modeling and Simulation (VAMS) project to develop new ATM concepts that can provide the needed increase in capacity. To evaluate these concepts, the developers are simulating them on the Airspace Concept Evaluation System (ACES) simulator. A single execution of ACES can require over 10 computers, running for many hours, simulating over 60,000 flights and generating tens of gigabytes of data. VAMS concept developers need visualization tools to understand the outputs of the ACES simulator and to better understand the strengths and weaknesses of their concepts revealed by the simulation data.

Plotting aircraft at their actual geographic locations is the standard approach to visualizing air traffic data, but this approach often results in displays that are cluttered and not very helpful in understanding the state of the NAS. Figure 1 shows two examples of this type of visualization developed by HRL. Seeing 5,000-10,000 aircraft plotted at their actual locations results in clutter and high densities. Aircraft can overlap each other, and there is insufficient space to plot much information on each aircraft other than their location, direction, and perhaps one or two more dimensions. National air traffic managers do not find it useful to display all aircraft; they instead focus their attention on smaller subsets. However, focusing on certain aircraft runs the risk of misinterpreting the state of the overall NAS due to seeing only part of the situation. More importantly, geographic plots do not usually reveal the subtle patterns, trends, correlations and relationships that are the insights that developers need to evaluate and improve their concepts. For example, a geographic display could show high densities in certain areas of the U.S., indicating that those areas are at capacity. However, it would not answer the question of what are the characteristics of those aircraft that are currently causing the congestion and how are they related to each other. Similarly, such a display does not answer the question of how delay is propagated through individual aircraft and the overall NAS, nor how delayed flights are correlated to key factors such as airports or geographic regions. Since geographic plots alone are insufficient, there is a need for more advanced visualization modes specifically to address the problem of generating insight from NAS simulation data.

Because of this need, the Air Force Research Laboratory funded the VisION project (Visualization for Insight into the Overall NAS). The intention of the VisION project is to develop new visualization concepts that do not rely primarily upon geographic plots, to aid VAMS concept developers in gaining insight from NAS simulation data.

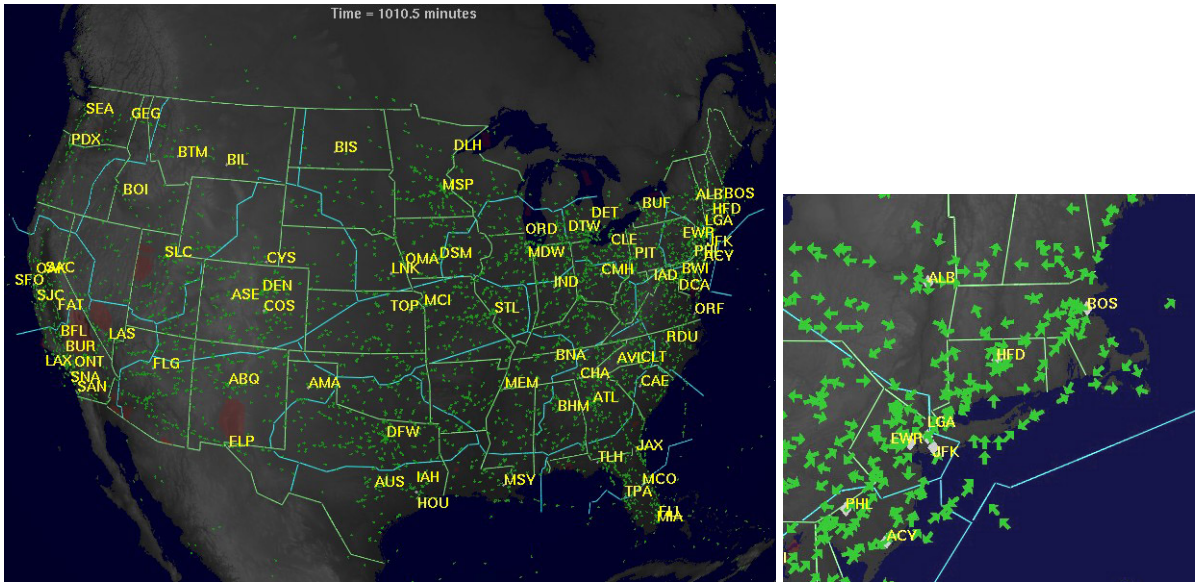


Figure 1. Examples of straightforward geographic plots of NAS traffic.

2.2 Goals and Requirements

AFRL primarily wanted us to generate ideas. Therefore, that is where most of our effort was directed. However, the only convincing way to demonstrate that a visualization concept has value is to eventually implement it and provide evidence of its worth through some sort of evaluation. Therefore, this contract also called for us to do preliminary implementation and evaluation steps.

We paraphrase the contract requirements in the following list, along with a brief description of where we document completing each requirement:

- *Provide 1 or more ACES datasets, with documented format* [There are five ACES datasets on the DVD-R, and the data formats are documented in the VisION software description document.]
- *Design 3 or more concepts* [We offer twelve concepts, which are described in this final report in Section 4.]
- *Prototype implementation of key aspects of one or more concepts* [We implemented three concepts. The source code is on the DVD-R. The description of each program is in the VisION software description document.]
- *Evaluate implemented concepts* [Our evaluation results are documented in the VisION evaluation report. Supporting movies and images are provided on the DVD-R.]

    - *Provide and document code* [The source code is on the DVD-R. The description of each program is in the VisION software description document.]

    - *Reports* [We provided all required monthly reports, the software and software description document, the evaluation report, and this final report.]

- *Meetings* [We held the kickoff meeting, ACES data transfer meeting, mid-project review (as a phone telecon at AFRL's request), and final presentation as called for in the contract.]

As a bonus, we also published a paper describing most of our visualization concept ideas. This paper was titled "Visualization Concepts for Generating Insight from NAS Simulation Data" and we presented it at the AIAA Modeling and Simulation 2005 Technical Conference, on August 18, 2005 in San Francisco.

To our knowledge, this is the first work offering visualization concepts specifically designed for extracting insight from NAS datasets, where the visualization modes do not rely primarily upon normal geographic plots.

3.0 IDEA GENERATION APPROACHES AND STRATEGIES

3.1 Idea Generation Approach

We began the idea generation phase by simultaneously reexamining the literature and holding brainstorming meetings to generate new ideas. The goal was to draw ideas from existing works while also affording the opportunity to generate a novel approach before being completely immersed in the previous visualization literature.

The most relevant publications were, not surprisingly, in the field of information visualization. This is the field of visualization techniques for viewing non-spatial data (or examining data in ways that do not involve rendering the data in spatially accurate plots). These include glyphs, scatterplots, aggregation strategies, treemaps, parallel coordinates, spiral graphs, network visualization, graphs, and others. We found several tutorials that survey the field and read those. We apply several of these techniques (and modify them) for our application.

The primary source of the information visualization literature is the IEEE Information Visualization conference, which we checked thoroughly. However, we found very few papers directly involving an ATM application. The most relevant one was a paper called “EdgeLens: An Interactive Method for Managing Edge Congestion in Graphs” by Wong, Carpendale, and Greenberg. This was published in Information Visualization 2003. This paper described the problem of maps that render many node-to-node connector lines, such as the maps showing all the routes that a large airline flies. When many routes go to a particular node (such as all of United’s routes to and from Chicago O’Hare), the density of route lines becomes too large to see the situation at that node. They offer methods of curving such lines away from a busy node, combined with transparency, to increase understanding of the connections at that node.

The more traditional visualization field is the scientific or data visualization field, which is generally concerned with visualizing spatial information such as flow fields (weather, vortices), medical data (volume rendering), architectural data, etc. This is represented by the IEEE Visualization conference, which we also examined thoroughly. We did not find these papers to be as relevant, although we do offer one concept based upon flow visualization.

Finally, we examined the air traffic literature, represented by the ATCA and AIAA conferences. All the previous visualization papers (which included our own papers) appeared to use normal, straightforward geographic plots. For example, NASA’s FACET project appears to generate 2D and 3D geographic plots that are similar to what AFRL has also demonstrated with JView. The FlightView product enables anyone to see real-time positions of most aircraft in the NAS, plotted on a map of the U.S.

To our knowledge, this project is the first to focus upon visualization of air traffic simulation data through methods that do not primarily rely upon geographic plots.

Our brainstorming sessions took place simultaneously and occurred in phases. We initially asked the HRL team members to bring up any ideas that they thought might be remotely relevant, with

the goal of "casting the net" as widely as possible. This included some highly speculative ideas, such as mapping the NAS state to anthropomorphic display modes (such as a face or human-shaped robot) that would give a gestalt impression of the overall NAS state. For example, a smoothly running NAS would appear as a happy person but a NAS with congestion and delay problems would appear as a sad person. We did not reject any ideas in this initial phase. After that, we sorted ideas into groups and began further developing the ideas that we thought had the most promise. Concurrently, we also examined the NAS at a more fundamental level and developed particular strategies based upon the characteristics of the NAS that we thought we could exploit. These are listed in more detail in Section (3.2). The ideas that we developed are documented in Section 4, while some others that we abandoned are listed in Section 3.3.

3.2 Visualization Approaches and Strategies

In this section, we discuss the high level approaches and strategies we took when designing the visualizations.

We note two inherent characteristics of this problem domain. The first is the size of the datasets. ACES produces a large number of data measurements. For example, a single execution can simulate between 10,000 and 100,000 individual flight segments. This is a large amount, too large to be easily readable on a geographic display, but it is not as large as some other types of data that have millions or billions of measurements (e.g., phone connections, web requests). In fact, our datasets are generally small enough to plot some information about each item so that the entire system-wide state is visible on one or two screens. For example, if we have two 1,280 × 1,024 displays, then that provides 2,621,440 pixels, or an average of 26 pixels per item if we have 100,000 items in the dataset. The second observation is that although our goal is to develop visualization techniques that do not primarily rely upon geographic plots, the geographic location of aircraft and other NAS elements is still very important. Therefore, our visualization approaches may use methods of abstracting or warping the geographic information, which will preserve some geographic information even if it is not absolutely accurate. Also, a basic strategy for some of our visualization modes is to use them as a filter and combine them with a normal geographic display. The role of the visualization mode is then to allow the user to qualitatively understand the overall NAS, then select a small subset of the NAS that the user wishes to explore. Then that small subset is what is displayed on the normal geographic display. This strategy avoids the clutter and density problems of plotting all the aircraft, since we only plot a small subset of the total aircraft.

In designing our visualizations, we also adopted a number of general strategies:

Develop NAS and ATM-specific modes. Most visualization techniques are designed for maximum generality. Because we are focused on this particular application area, we often took the opposite approach. We designed custom visualizations intended to answer specific questions in this problem domain or to look at characteristics specific to the NAS and the ATM domain. In particular, we called the ATM-specific features the "points of leverage," shown in Table 1. The ones that we made use of are listed in italics. Other characteristics, such as weather features and individual passengers, are also important but are not simulated in ACES and therefore data is not

available. Table 2 lists the metrics that ATM personnel use to measure the performance of capacity-enhancing concepts. Of these, we focused on congestion, capacity and delay, since those are the primary measurements that are available in ACES. The other factors are either not available or only simulated to a limited degree.

Table 1. Points of leverage in the NAS and ATM problem domain.

- *Airports*
- *Aircraft and flight crews (pilots, dispatchers)*
- *Flight routes (airport-airport pairs, segments)*
- *En route zones, sectors, TRACONs and other spatial regions*
- *Categories of delay*
- Weather features
- Passengers
- Money (payload: passengers and cargo)
- Airlines

Table 2. Metrics.

- *Congestion*
- *Capacity*
- *Delay*
- Safety
- Cost
- Environmental impact
- Customer satisfaction
- Workload

Define what the output should be. Humans are very good at detecting certain types of changes in images. We therefore design visualization modes to take advantage of these abilities. One method to achieve this is to define what the ideal output should be, in a manner such that deviations from the ideal appear as visualization cues that the user can detect preattentively. For example, it is known that humans can perceive even small differences in the orientation of adjacent vertical lines. If we define the ideal state (where no problems exist) as one where all the lines are drawn vertically, then we can define problems (congestion, delay) as turning the orientation of those lines away from vertical. Then a user can tell, at a glance, if problems exist and which items are affected.

Self-organization and optimization. In some of our visualization modes, we need to plot representations in some non-overlapping order or array. This leads to the question of how to arrange these plots: in what order should we render them? One general strategy for solving this problem is to define the characteristics we want the visualization to achieve, then use self-organization and optimization algorithms to automatically determine the order of placement. This is generally done by defining a cost function. For example, assume the visualization should cluster similar entities together so that patterns become more salient. Then the cost function rewards putting similar items next to each other and penalizes putting different items next to each other. The optimizer then searches for an ordering that minimizes the cost function.

High-resolution display. The amount of detail (number of pixels) that we can assign to any item depends on the number of pixels available. A typical PC, driving two screens, only provides about 2.5 million pixels. Higher-resolution displays and systems do exist, but they are not common. An alternate approach for prototyping visualizations that require high resolutions is to print the rendered outputs rather than displaying them on a monitor. The resolution achievable in print is much higher than with most monitors. However, a problem with this approach is that the user cannot easily interact with the visualizations. Despite that, printing is a useful strategy for exploring high-resolution visualization prototypes at low cost.

Ask the users what they need. Finally, we asked the VAMS concept developers what they wanted to discover using visualization tools, or what questions they needed to ask about their concepts where a visualization tool might help them. To some extent, we cannot expect the users to be able to describe what they really need because a successful visualization tool will help them discover answers to questions that they didn't even know to ask in the first place. In general, we want our visualization tools to identify correlations between various combinations of the "points of leverage" and the metrics, to make outliers, exceptional cases and interesting patterns salient. However, the VAMS concept developers did provide some guidance on more specific questions of interest. We did not pursue some of the questions that they provided, because some were outside the scope of this project (Prove that Free Flight is better than controlled flight) or too specific and answerable as a database query (Show all the flights into airports with congestion above a certain threshold). The particular questions that they proposed that were most relevant and within the scope of what we can address using ACES simulation data are:

- How are resources (airports, sectors, etc.) overloaded with time throughout the NAS?
- How are delay components correlated with other aspects (airports, geography, etc.)?

- How is delay propagated according to flight segment?
- For all delayed flights, what are their routes (airport pairs) and correlations to airports?

Because of this guidance, we developed some of our visualization concepts to address these particular questions.

3.3 Ideas Considered But Abandoned

In this section, we briefly discuss some ideas that we considered but abandoned or chose not to further develop for various reasons. We document them here to show that we considered an even wider variety of concepts than the final list of 12 that we settled upon.

Some of these are graphically illustrated in the backup slides for the VisION mid-project review, in the Powerpoint presentation file "VisION_midProject_bkp.ppt."

Head-mounted displays. We briefly considered using Head-Mounted Displays in Virtual Environment type of applications, since that invokes the user's spatial memory and can provide a large, 3D environment to render information inside. However, display resolution is still quite low in any HMDs that are reasonably priced, and since high resolution was vital for most of the concepts we considered, we quickly dropped the idea of using an HMD.

Plumbing analogy and network/train visualizations. One of the first approaches we considered was a plumbing analogy. This is very similar to visualizations of networks, train station and rail lines, and subway stations and routes. We can consider the airports to be the nodes or stations, while pipes connecting the nodes are the various flights. The width of the pipe corresponds to the capacity, while the aircraft themselves are the "water." This analogy is somewhat appropriate for overseas routes, where capacity is directly linked to the routes themselves since they do not have ground-based radars to track aircraft over the ocean. However, this analogy breaks down in domestic airspace, because capacity is tied to particular areas of airspace (e.g., sectors) rather than particular routes.

3D Theme River. Harve, Hetzler and Nowell introduced a visualization concept called "Theme River" that displayed different categories of data plotted against time, where the different categories appeared as layers in an object, as if deposited in a sedimentary process. This could be applied to our application by showing different amounts of aggregated delay and congestion at different times. We were initially attracted to this because the overall size of the layers represents the total amount of a metric represented, and this could show the natural increase and decrease in delay or congestion that occurs as a day progresses in the NAS. However, this is still an aggregated display and we felt that other visualization concepts we did further develop would provide more detailed information than this one.

Audio. It has been demonstrated that blind people are able to "see" coarse objects through a visualization technique that maps a low resolution (e.g., 64×64) grayscale image of the world into a brief sound. For details, see the web site (www.seeingwithsound.com). Significant user training is required, but this sonic visualization has enabled blind users equipped with an

inexpensive wearable camera to detect doors and obstacles in front of them. We were initially interested because multimodal techniques could enable users to understand more complicated data, and it is known that skilled sonar operators can detect objects even in noisy and confusing audio signals. However, this sonic visualization requires mapping many dimensions into a single 1D signal, and it requires considerable effort to interpret. The user's understanding is also low resolution: the user can determine that a door is roughly in front of him or her, but cannot discriminate between a door two inches to the right versus straight ahead. Since our users are not blind, it made more sense to concentrate on the highest bandwidth information display channel: the visual sense. However, audio still might make sense when used in conjunction with other sensory modes.

Haptic. The NAS has been compared to a fluid where the aircraft are water particles that have minds of their own. As the situation changes, the fluid moves and responds in ways that can be complicated to understand; simple rules like "water flows downhill" are insufficient. We considered modeling the NAS as a "haptic water" surface, where the height of the surface represented delay or congestion, and the user could press upon particular points with a haptic force-feedback device (like the Phantom arm from SensAble Technologies) and then feel the resistance and observe how the fluid changes in response to that pressure. Areas with spare capacity would then feel soft and pliant, but areas with no spare capacity would offer great resistance. However, implementing this would require a large number of underlying simulation runs that represent the different conditions away from the base condition, and we did not have such ACES datasets available. Also, commonly available commercial haptic displays would allow the user to probe the surface at only one point at a time.

Node and graph displays. There are a variety of node and spring, hyperbolic and other displays that automatically reposition nodes (such as aircraft) and links (such as flights between airports). Therefore those could be worth considering for viewing flights between airports. However, such displays generally get cluttered quickly and only allow a user to see a fairly small number of objects (in the hundreds) before they become unreadable. Since we want to see many more objects and relationships simultaneously, we did not think this was the best strategy for our application.

Causality displays. The visualization literature offers many ways of displaying hierarchical data. The classic example of hierarchical data is a computer's file directory, and methods for displaying these include cone maps and treemaps. This would be a natural way of showing causal relationships. For example, if event X caused a set of consequences Y, and those in turn generated consequences Z, then we could show how all of these were linked through a hierarchical visualization. However, ACES does not directly provide any information about cause and effect. We can see correlations, but the user has to infer causality, which is a much harder thing to determine and verify. Therefore we have to leave causal relations to the user's interpretation.

Passenger-focused displays. We briefly considered the idea of generating visualizations that focus on the passengers in the NAS. In 2004 there were 619 million passengers, or an average of

1.7 million per day. If we displayed the condition of these passengers, we could see the overall state of the NAS from the perspective of the paying customer. In particular, airlines might find this useful at hub airports, where planners could understand the ramifications of holding aircraft to allow delayed passengers to complete connections, versus sending aircraft out on time but having to reschedule or strand some passengers. The main problem is that we have no data on individual passengers. This is only available to the airlines themselves, and they keep this secret since this would be useful to competitors. Since we had no data, we did not attempt to develop displays along this direction.

4.0 VISUALIZATION CONCEPTS

In this section, we describe each of the twelve visualization concept ideas we presented at the mid-project review. Each is described in its own subsection.

4.1 Multiscale Geospatial Classification

One question requested from the concept developers was a means of showing how delay components are correlated with other aspects of the NAS. The multiscale geospatial classification visualization concept is intended to address this request. Many different factors can contribute to the overall delay of a particular flight. The aircraft may be held at the gate, or delayed during the taxi to the runway, prevented from taking off, rerouted or held during the en route phase of flight, and similarly delayed during arrival on the taxiways and gate of the destination airport. The total delay for all flights within particular regions (zones, sectors, airports, etc.) can be aggregated and then represented by icons of different patterns, colors and sizes, where the size is correlated to the magnitude of that type of delay. These icons can then be positioned at their approximate geographic locations (Figure 2).

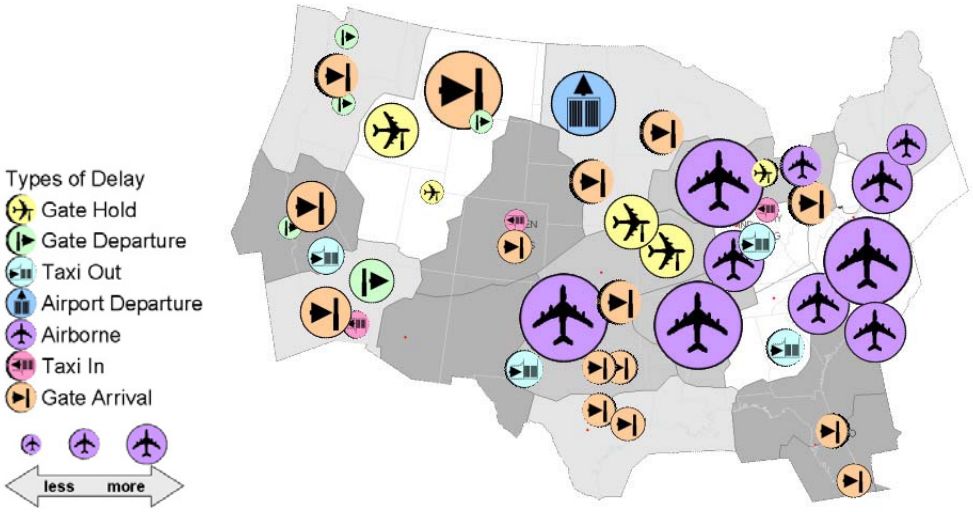


Figure 2. Multiscale geospatial classification image.

This visualization leverages strengths in visual perception to reveal large scale patterns of delay. The user should be able to tell, at a glance, the qualitative situation of the delay problem and to recognize particular patterns between delay and geographic regions. Visual grouping enables the observer to detect which regions share similar types of delay problems. The apparent spatial frequency and visual density can indicate regional effects resulting from that type of delay. The relative scales of the icons indicate the relative magnitudes of the delay problems.

For example, in Figure 2, the user can quickly see that large amounts of airborne delay exist in the Midwest and East Coast, along with many cases of arrival delays at airports spread along the western and southern U.S. From this correlation, the observer may infer a causal relationship: that arrival problems at key airports in the west and south are causing delays of in-bound flights in the Midwest and East Coast.

4.2 Delay/Congestion Correlation

Although this concept shows correlations between delay and congestion across the NAS, it is more generally a method for displaying glyphs representing any aspect of the NAS, where the glyphs are tied to individual sectors in the NAS and are rendered in a way that avoids overlaps. It remaps straightforward geographical plots to a more abstract representation, based upon the existing NAS hierarchy. The NAS is divided into 20 "En route" zones, covering the continental U.S. These "En route" zones are large regions of space with names like ZLA (around Los Angeles) and ZFW (around Dallas-Fort Worth). We can map these 20 en routes zones into a 5x4 array (see Figure 3). For a 1,280 × 1,024 display, we can allocate 250 × 250 pixels to each of the squares in this array.

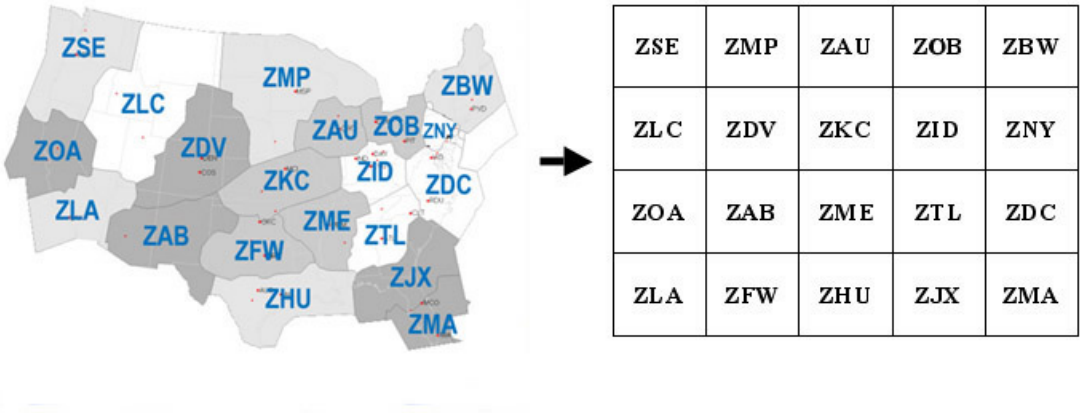


Figure 3. Remapping En Route zones to 5 by 4 array.

Each "En route" zone is divided into an average of 40 sectors. These sectors are grouped into three categories of altitude: Low, High and Super sectors (listed in ascending order). For each "En route" zone square, we divide it into those three altitude groups and dedicate a box for each sector. In each box, we can render a glyph, showing congestion in orange and delay in blue (Figure 4). While this is a particular glyph to show the correlation between delay and congestion, this visualization concept could use any other glyph design to show other desired NAS properties in each sector.

Figure 5 shows an example of what an overall visualization might look like. Areas that do not have congestion or delay problems (above a certain threshold) are drawn as empty regions, so the observer can immediately tell which regions are not a problem. Areas with congestion have orange bars and areas with delays have blue bars. Areas where the two are correlated have both colors. The user can quickly tell the approximate geographic region where these problems occur, what type of sectors (altitudes) these occur in, and the relative magnitude of these problems.

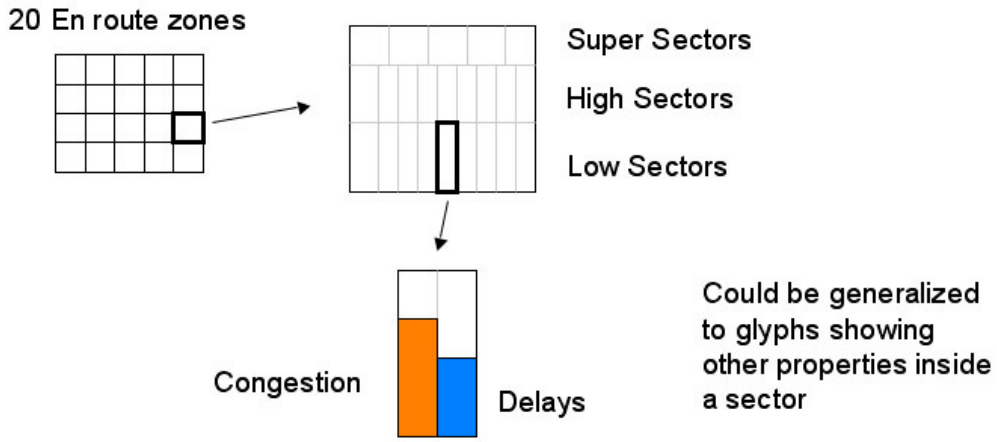


Figure 4. Glyphs for individual sectors.

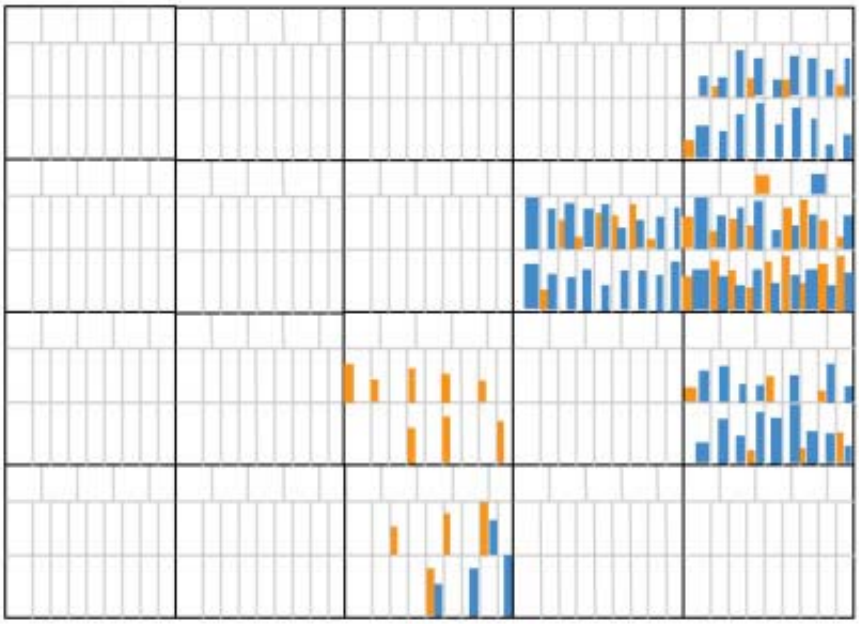


Figure 5. Visualization concept for delay/congestion correlation.

4.3 Self-organizing Scatterplot

The Self-organizing scatterplot visualization concept was developed to address one of the questions submitted by the concept developers. Specifically, it visualizes correlations between delayed flights and the airports those flights use. A geographic plot can identify which aircraft are delayed and where those aircraft are. However, it is not trivial to then show which airports are associated with the delayed aircraft. Drawing route lines to identify the airports quickly results in a cluttered display. Instead, we can take a scatterplot approach. Each aircraft has a

departure and an arrival airport. We can use an XY plot, where the departure airports are listed on the X axis, and the arrival airports on the Y axis, and the airports are listed in the same order on both axes. Then at each intersection, we render a glyph. Off-diagonal intersections represent delay along a particular flight route, such as LAX to ORD. Figure 6 shows an example of a glyph design for these off-diagonal intersections, which encodes total delay, components of delay, and the approximate geographic locations of the flights involved along that route. Then the diagonal intersections represent characteristics of particular airports. We use a perceptually different glyph here: a star glyph, where lines are draw in the direction of the departing and arriving flights, where blue means departure and green means arrival. This shows the directions and magnitudes of all delayed arriving and departing flights.

Finally, there is a question of what order to list the airports. The goal is to put the airports in an order that makes patterns and clusters obvious. For example, areas with similar delay patterns should be plotted next to each other. Given N airports, there are N! possible orderings. This is too many for us to try all possibilities. Instead, we take the self-organization strategy and define a cost function that rewards putting routes with similar delay values adjacent to each other. Then we can use an optimizer, such as Adaptive Simulated Annealing to find the best ordering.

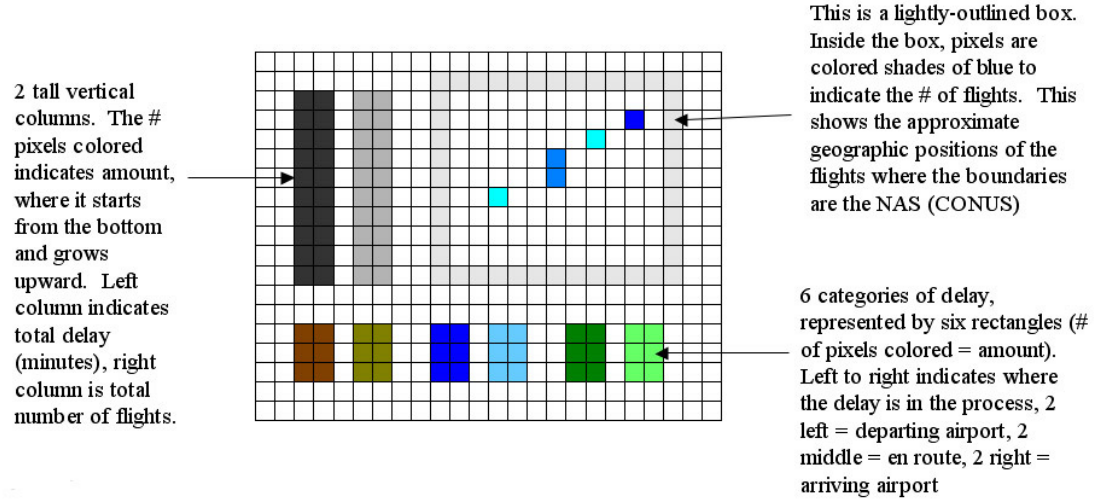


Figure 6. Delay glyph design for self-organizing scatterplot.

Figure 7 shows an example of what a scatterplot might look like, for a simplified case involving only ten airports. Particular types of delay problems appear as recognizable patterns. Horizontal groups appear because of problems with a departure airport. Vertical groups appear because of problems with an arrival airport. Clusters indicate more complicated problems involving geographic regions. For example, in Figure 7 we can see arrival problems into Boston from some airports along the East Coast, and a cluster of problems from three West Coast airports into three East Coast airports.

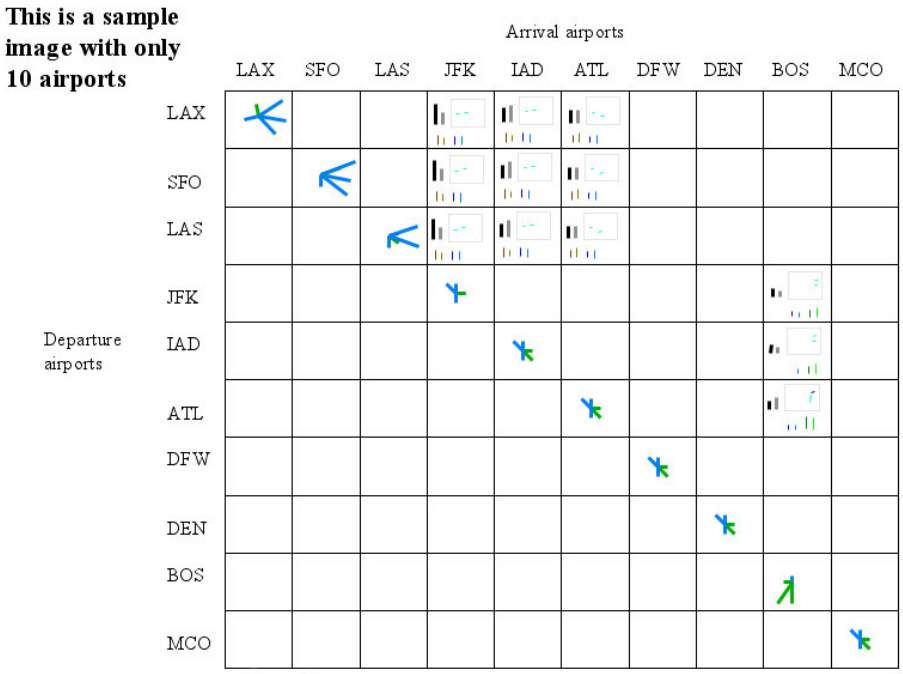


Figure 7. Visualization concept for the self-organizing scatterplot.

4.4 Airport – Airspace Correlation

The Airport – Airspace correlation concept was developed specifically to address one of the concept developer requests: "How are delay components correlated with other aspects (airports, geography, etc.)?" More specifically, if we have an area of airspace that is congested or has many delayed flights, what is the relationship between that geographical region and the departure/arrival airports associated with each flight in that airspace?

To visualize this relationship, we use an X-Y graph where the Y axis represents the top 250 airports in the U.S. and the X axis represents all the sectors. Both the airports and the sectors are grouped into the 20 en route zones covering the continental U.S. These zones are listed in approximate west to east order, and this order is the same on both the X axis and the Y axis. On the Y axis, within each zone, airports are sorted so the largest (in terms of maximum number of operations per hour) are at the top and the smallest are at the bottom. Similarly, on the X axis, within each zone, sectors are listed in order of altitude: low sectors on the left, high sectors in the middle, and super sectors on the right. We plot either delay or congestion (but not both simultaneously). For each sector, we compute the congestion (number of aircraft versus the maximum sector capacity) or the total amount of delay experienced by aircraft currently in that sector. If the congestion or delay is above a user-specified threshold, we plot all the aircraft in that sector. One sector occupies a column in the graph. Each aircraft has an associated departure and arrival airport. Therefore we tally the number of aircraft departing from and arriving at each airport in that column. When that count is completed, we render the column, drawing arrival airports in red and departure airports in blue. Airports that have both arriving and departing

flights become purple. The brightness of the pixels varies with the number of aircraft assigned to that airport.

Salient features in this visualization will indicate specific types of problems. For example, a horizontal line indicates a problem with a particular airport, and the area that the line covers specifies the region of the NAS where the problems are occurring. Similarly, a vertical line indicates a problem with a particular sector and the extent of the line identifies the arrival and destination airports affected by this problem. Features along the main diagonal (from the upper left corner to the lower right corner) indicate mostly local problems, where the affected airports are geographically close to where the flights are currently located. Clusters indicate more complicated problems affecting particular regions.

Figure 8 shows an image from a preliminary implementation of this concept. This renders delay for a 2×ORD ACES dataset. In this particular dataset, all aircraft either depart from or arrive at Chicago O'Hare (ORD). However, if the user was not originally aware of the unusual nature of this test dataset, this visualization would make it clear that the problems are concentrated at ORD. The vertical red line is in a sector in the ZAU en route region where ORD is located. This vertical red line indicates a large number of delayed aircraft near, or actually still at, ORD that need to arrive at other airports spread across the NAS. There is also a horizontal blue line for the first airport in the ZAU en route zone (which is ORD). This indicates aircraft that have already departed from ORD and need to arrive at other airports. Those destination airports are indicated by the red pixels, spread roughly along the main diagonal. Since those red pixels are mostly along the main diagonal, that indicates that these airborne flights are getting close to their arrival airports. The overall conclusion is that the delay problems are due to aircraft departing from ORD.

Figure 9 is another image from our preliminary implementation, this time rendering congestion for a 2×NAS dataset. This plots aircraft in congested sectors, where a sector is considered to be congested if the total number of aircraft in that sector is >90% of its maximum capacity. There are several notable features in this image. First, it is interesting to see what is *not* plotted: the columns for the en route zone ZFW (around Dallas) are empty, indicating a lack of congestion problems in that entire zone. Next, there is a horizontal red line for an airport in ZBW and a horizontal blue line for the top airport in ZME. These reveal congestion problems that apparently link these two airports together. These problems are spread across much of the NAS. (This is probably a bug or an artifact of the data; see the evaluation report for further discussion.) The purple features along the main diagonal indicate local congestion problems, where there is congestion in the airspace around an airport due to aircraft either arriving at that airport or departing from that airport. We note that the density of pixels tends to be higher for the airports at the top of each zone on the Y axis, compared against the bottom of each zone. This suggests that congestion problems tend to be associated with the largest airports, rather than the smaller airports. Finally, there is a higher density of pixels in the lower right quadrant than in the other three quadrants. This suggests that there is more congestion in the eastern and northeastern U.S., compared to other regions, and that much of this congestion is due to flights that begin and end within the eastern and northeastern U.S.

For more discussion and evaluation of this concept, please consult the VisION evaluation report.

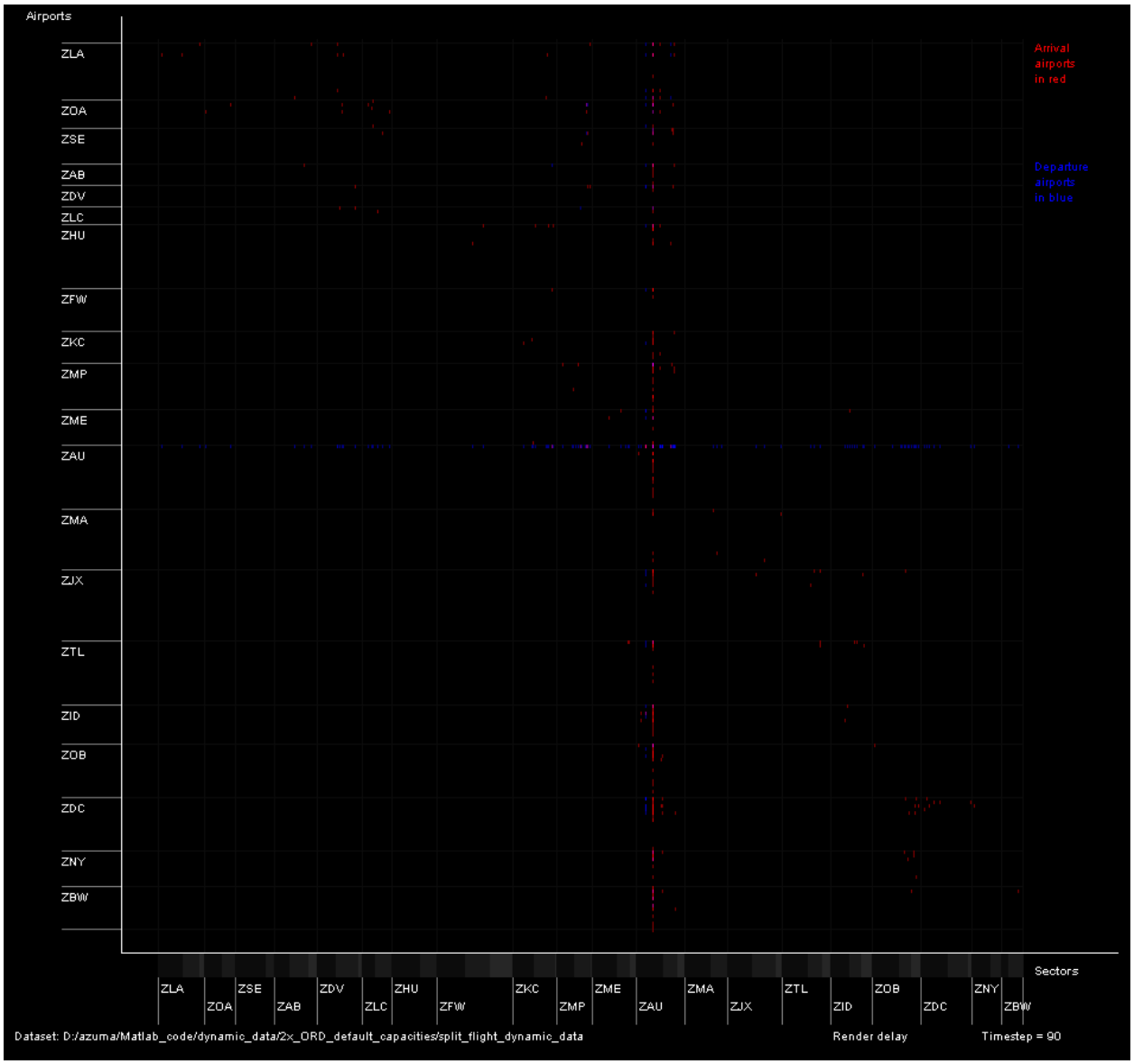


Figure 8. Airport – Airspace correlation implementation, rendering delay for a 2×ORD ACES dataset.

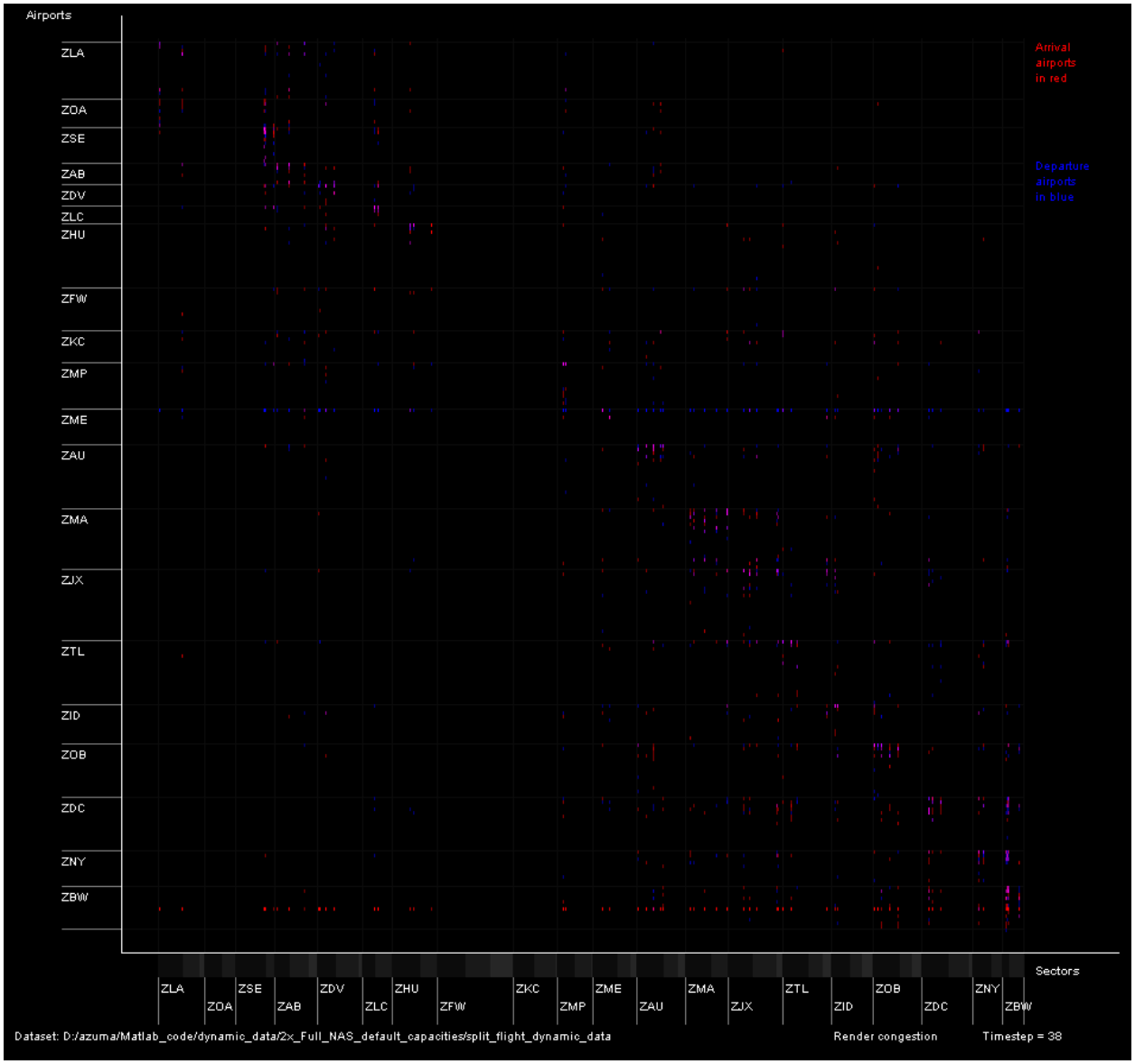


Figure 9. Airport – Airspace correlation implementation, showing congested sectors (>90% capacity) for a 2×NAS dataset.

4.5 Transformation to a Different Basis Space

In many cases, concept developers will want to compare the results of one simulation run against another. For example, there might be a base simulation run that represents the performance under today's ATM system. Then they might compare the simulation results of one concept against that base, or compare different capacity-enhancing concepts against each other when presented with the same situation. Currently, ACES simulation runs have been performed on selected days (such as May 17, 2002: a high-demand day with few weather features) to provide a basis for comparing concepts in this manner.

To support such comparisons, we would like to develop a visualization mode that makes it easy to interpret the differences between one simulation run and another. This comparison should take into account how the NAS evolves with time, rather than only comparing the situation at one particular point in time.

Our approach was inspired by signal processing techniques of converting time-domain signals into the frequency domain, through Fourier, Laplace, or other transforms. Once plotted in the frequency domain, certain properties of the signal (spread out over time) become obvious and certain calculations are easier to perform. We considered simply applying a 2D transformation to the geographic locations of the aircraft, but it was not obvious that would achieve the results we desired. Instead, we propose a hierarchical method of describing differences from a NAS "basis function."

First, assume that the item of interest is congestion in sectors. The output of the simulation run reports the congestion in each of the ~800 sectors at each timestep (for example, every 15 minutes). We can then define a NAS "basis function," that given a sector number and a timestep number, returns the amount of congestion. This defines the performance of the NAS in this simulation run for one 24-hour day.

Now we want to compare a different simulation run against this basis. Assume, for the sake of argument, that the new simulation run had exactly the same congestion as the basis. Then we can describe the new simulation run as being the basis function with a coefficient of 1. We only need one number to convey the essential property – that the two simulation runs generate the same outputs, both in space and time.

However, the new simulation run will usually be different in some fashion. To describe this, we start with the basis function (assume this is the May 17 simulation run) and then add other functions that affect the output in a hierarchical fashion. Figure 10 shows the idea. We have two levels of hierarchies (although this could be divided into other finer resolutions if desired), both in space and time. The first, coarser level divides space into en route zones and time into morning, afternoon and evening. For example, one function would be for en route Zone #1 in the morning, and returns the amount of congestion that when added to the basis function for all sectors in that en route zone and for all morning timestamps, minimizes the difference between the basis function and the new simulation run. Of course, this alone will not usually precisely match the congestion in the new simulation run. Therefore we need a second level of correction functions, where each value adjusts the congestion level for a particular sector and a particular timestamp. These are the arrays of squares at the bottom of Figure 10.

Figure 11 shows what this visualization might look like for one example. It enables a user to quickly see how one simulation run differs from another basis, both in space and time. For example, in this figure, we can see that the new simulation run has more congestion in en route Zone #2 during the evening hours. This concept makes it feasible to describe the main differences between one simulation run and another through only a small number of coefficients.

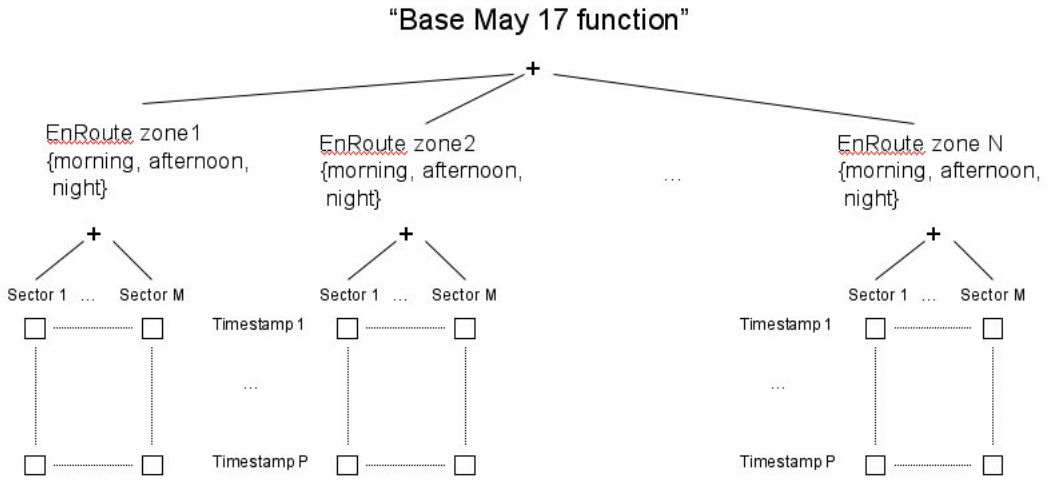


Figure 10. Basis and correction functions in the transformation concept.

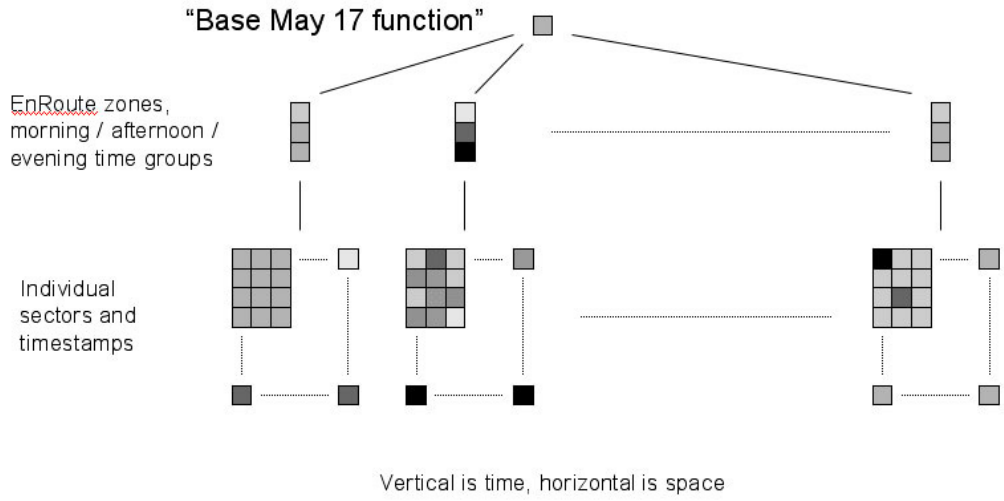


Figure 11. Visualization concept for the transformation to a different basis space.

4.6 Space/Time Extrusion

The idea behind this visualization concept is very simple. Most visualizations of a NAS state at a particular point in time are 2D plots. We can show the NAS state across many points in time by extruding this 2D plot into a third dimension. For example, if the 2D plot occupies the X and Y axes, then we extrude along a Z axis that is perpendicular to X and Y. While the plots might be geographic in nature, they could also be more abstracted measurements. The hypothesis is that people would be able to recognize particular 3D shapes inside the extruded volume and correlate those shapes with particular problems in the NAS. The left part of Figure 12 shows a very rough artistic conception of shapes inside an extruded volume. For example, if there are congestion

problems tied to a particular airport, then one might see an ellipsoidal shape, where the ends represent the start and stop times of the congestion problem and the extent of the ellipsoid reveals the area affected by the congestion problem. The right part of Figure 12 shows an example of what shapes one would expect on an individual flight. We would expect to see a line or bent cylinder that traced out the flight path in time. Weather features or other problems might be represented as 3D obstacles in the space-time (volumes that aircraft are not allowed to enter). Adjustments around that 3D obstacle would appear as changed slopes and directions of that line or bent cylinder.

Implementing this concept would require the application of volume rendering techniques, such as splatting, using cut planes, or computing and rendering isosurfaces.

The hypothesis of this concept is that users might understand the nature of NAS problems as they progress in time by recognizing and interpreting 3D shapes.

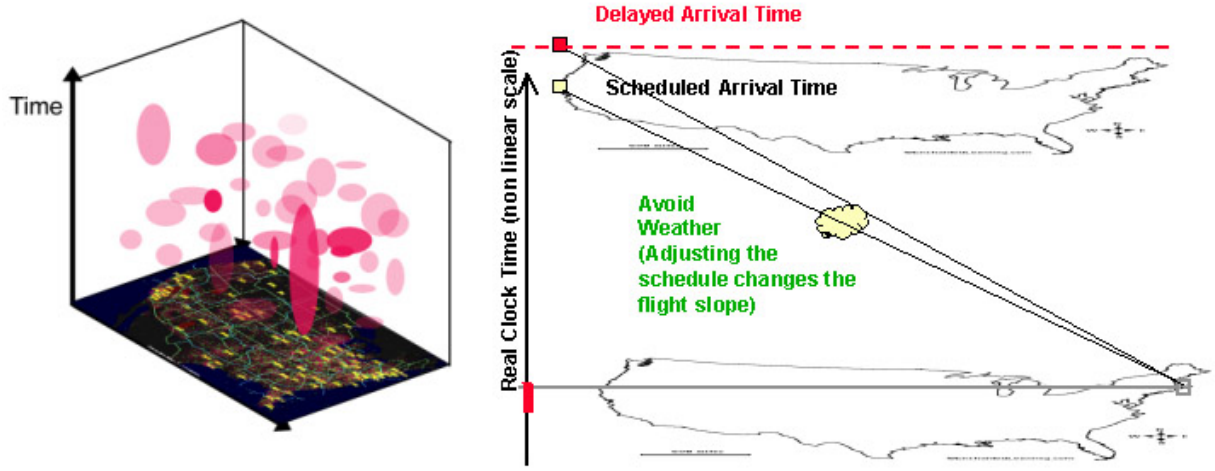


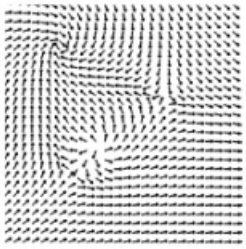
Figure 12. (Left) Artistic conception of space-time extrusion visualization concept. (Right) Example of an extruded shape for a single aircraft, and how that shape changes to reroute that aircraft around an obstacle (which itself is a 3D feature in space-time).

4.7 Flow Visualization

Flow visualization is a traditional area of scientific visualization research. Since there is a large body of literature of inventive methods for visualizing flow fields, we thought that there might be a way to map the NAS state into a flow field, and then use traditional rendering methods upon that mapped data.

However, we have a problem in that flow fields require even spacing between grid points, but NAS objects are not typically evenly spaced. All traditional flow field visualizations assume we have samples of what the flow is along an evenly spaced 2D or 3D grid, such as the image on the left part of Figure 13. However, NAS elements such as aircraft or spatial regions are not evenly spaced (as shown by the right part of Figure 13). If we mapped flow directly to the actual spatial locations of aircraft, then flow would only be defined at the points where an aircraft

existed. Furthermore, flow could be ill-defined at some of those points because one could have multiple aircraft at the same 2D coordinate but at different altitudes, so the flow at that point could be contradictory.



Flow field:
even spacing

Aircraft:
uneven spacing

Figure 13. (Left) Even grid spacing in flow field. (Right) Uneven spacing of aircraft in NAS.

Instead, we need to map NAS measurements to an evenly spaced grid. There are two approaches. We can take a direct approach, which samples the closest geographic values and generates an evenly spaced grid from that. Or we can take a more abstract approach, in which we define what a flow field should be based upon NAS simulation outputs. For these two examples, we'll assume we plot delay.

For the direct mapping approach, we map the aggregated delay of all aircraft onto an evenly spaced grid (Figure 14). We do that for a series of discrete timesteps. Then we can compute flow by taking numeric differences of how the grid values change with time (Figure 15). This is directly analogous to computing optical flow fields in computer vision. If we compute these around particular features in the NAS, such as a thunderstorm or other weather feature, we can infer the progression of this feature by looking at the magnitude and direction of the flow. For example, Figure 16 gives an example of a resultant flow field around a moving thunderstorm, and from the direction and magnitude the user might infer that the effects of this storm are getting smaller with time.

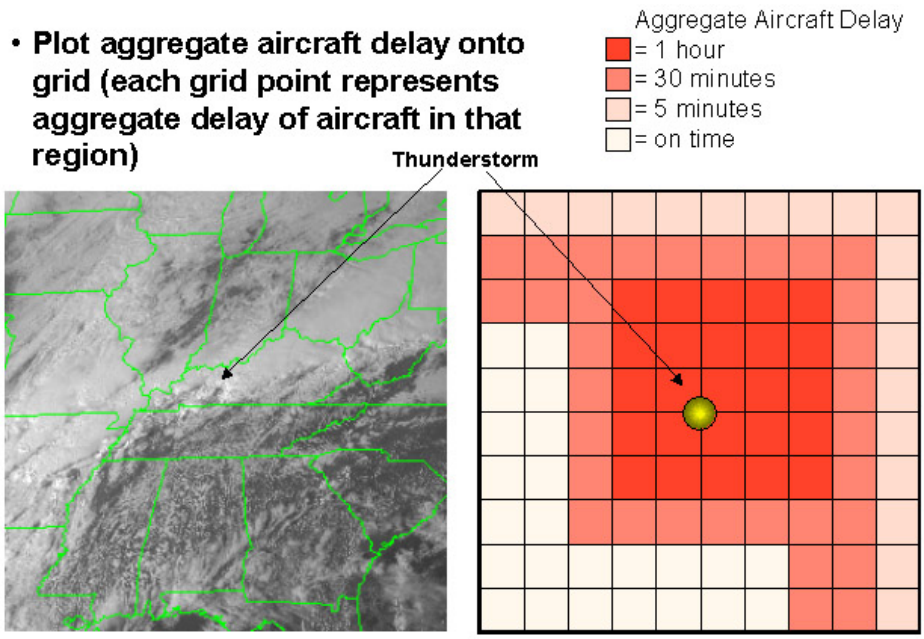


Figure 14. Mapping aggregate aircraft delay onto evenly spaced grid.

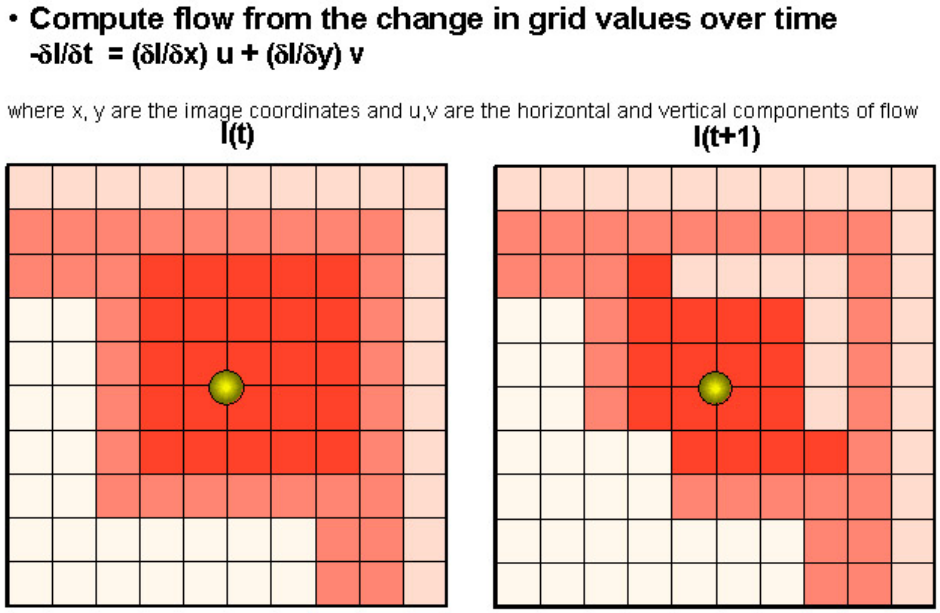


Figure 15. Computing flow from the change of grid values over time.

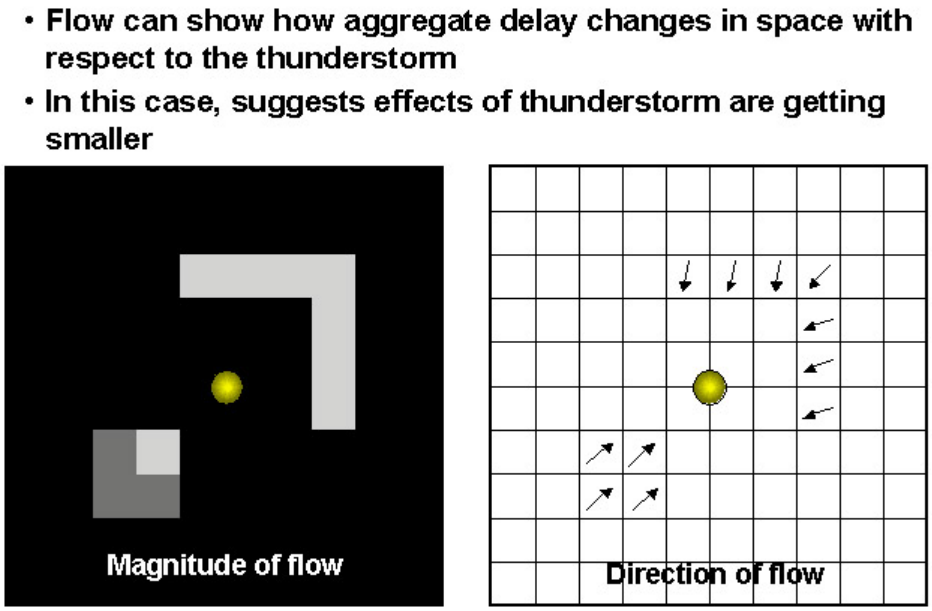


Figure 16. Magnitude and direction of flow field around a weather object.

For the abstract approach, we define a flow vector field to show the delay characteristics for a set of aircraft. Each aircraft is a downward facing line. The flight starts at a horizontal start line and the point where it is scheduled to currently be at is a horizontal line below the start line. We draw all the aircraft as adjacent lines. Figure 17 shows examples of different types of delay characteristics. A flight that is perfectly on schedule is a straight line that ends exactly at the lower horizontal line. An aircraft that is ahead of schedule extends below the lower horizontal line, while one that is delayed ends above the lower horizontal line. We can also represent delay by elevating the line in the third dimension. Aircraft that are forced into holding patterns could be represented by reversals of course and loops in the flow line. Aircraft that are put on hold, either at the departure airport or the arrival airport, could have special coloration to indicate that difference in status.

Then the next step is to arrange all aircraft of interest into some logical order. For example, if this were the case of examining all aircraft arriving at a particular airport, we could arrange the flights in order of arrival, from left to right. The spacing between lines indicates the temporal spacing between arriving flights. This visualization could reveal any problems in the flow of arriving flights. Figure 18 provides an example. On the left, we see a number of aircraft that are on schedule or a little ahead of schedule. However, in the middle there is one aircraft that is delayed and several aircraft after that that are in holding patterns. After that, the density of arriving aircraft decreases and the following aircraft are still on schedule, indicating that the delay problems have dissipated due to the lowed rate of arrivals.

For an NAS-wide visualization, we would have to seek a different grouping and ordering of aircraft that would allow patterns to arise in ways that are easy to interpret.

In conclusion, we have suggested two ways that standard flow visualization techniques could be applied to the NAS visualization application in ways to understand delay problems in the NAS.

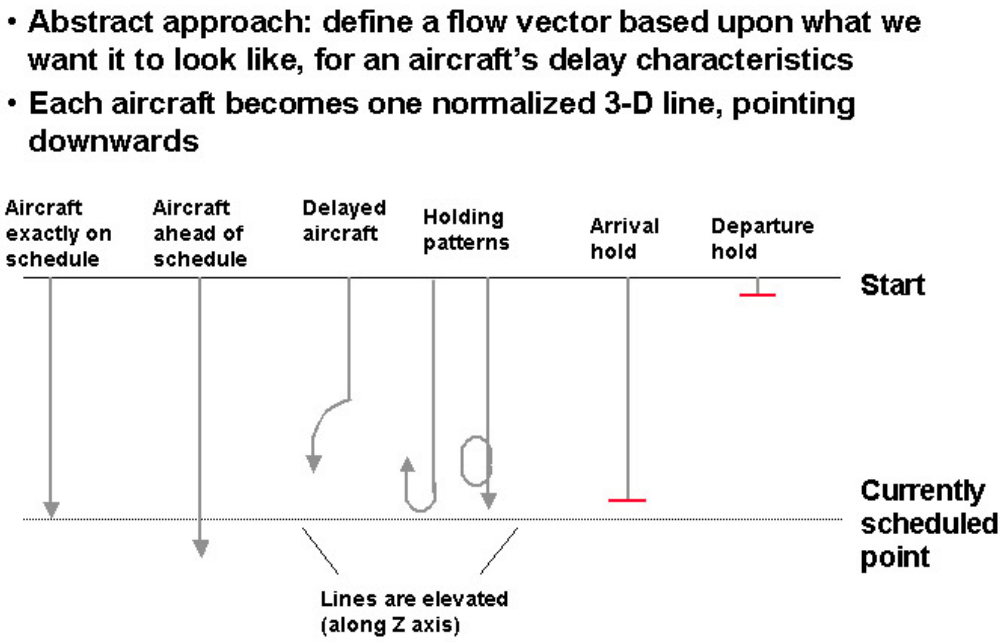


Figure 17. Examples of abstract mapping of aircraft delay characteristics to a flow field.

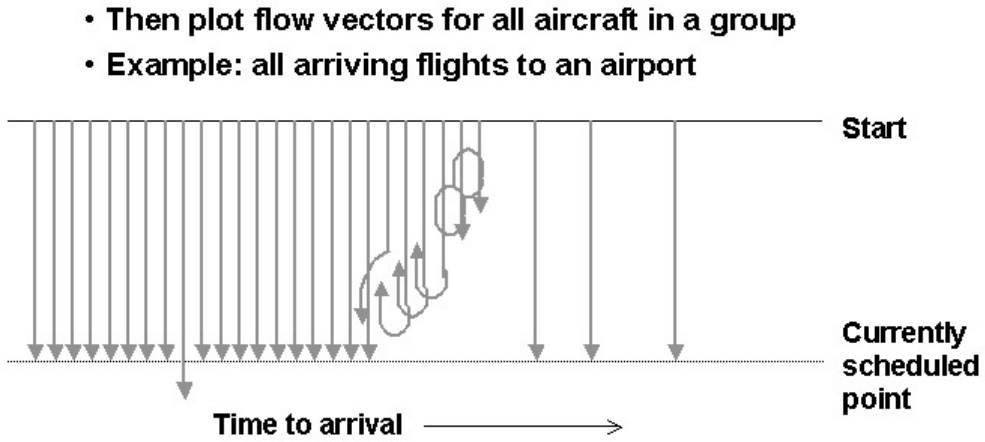


Figure 18. Example of abstract flow visualization concept of a stream of aircraft arriving at an airport.

4.8 GeoSPI Delay Propagation

If 60,000 flights occur in the NAS, that does not mean that 60,000 actual aircraft flew that day. One aircraft could account for multiple individual flight segments. For example, a Southwest aircraft might fly from Los Angeles to Las Vegas to Seattle to Portland to Los Angeles, all in one day. Thus one physical aircraft (identified by a unique tail number) accounted for four flight segments. Therefore, these flight segments are linked together by this physical aircraft. If the aircraft is delayed or sidelined, then all subsequent flights relying upon that aircraft will be affected.

One of the concept developer questions focused on this link, asking how delay is propagated according to flight segment. This visualization concept is intended to address this question. We first note that there are two dimensions that we wish to convey: the temporal aspects (the amount and type of delay) and the spatial aspects (the flight segment routes). Figure 19 summarizes these two aspects. For the temporal aspects, we use an hourglass representation where the upper triangle indicates the scheduled time and the lower triangle indicates the actual time. If the two meet (into a perceptually recognizable hourglass shape), then the flight was exactly on time at that point. If the two are separated, then the separation (highlighted in grey) represents the delay in achieving that particular milestone.

We apply a concept we call Geospatial Semantic Preservation for Interpretation (GeoSPI), which blends the temporal and spatial into one unified display by relaxing constraints imposed by the geospatial representation. Instead of plotting items on a completely accurate geographical map, we modify the spatial representations so they still retain their basic shapes (so they are recognizable by an ATC expert) but they are drawn at different scales and positions so that details are visible (for example, routes and delays within an airport, which would be invisible on a national map).

We provide two examples of GeoSPI visualizations that show flight segments for a single physical aircraft. In Figure 20, the arrangement emphasizes the scale of flight operations, with the national scale at the top, the sectors in the middle, and the airports themselves at the bottom. The overall delays are shown on the timeline at the very bottom.

In the second variation, we emphasize the individual flight segments themselves. This shows the first flight segment (from San Francisco to Los Angeles) on the top half, and the second flight segment (from Los Angeles to Cleveland) on the bottom half, with the timeline in the middle. This variation makes the flights segments themselves more obvious, while still permitting the user to see both the types of delay and the physical locations where those delays occur, all in one understandable diagram. Figure 21 shows this variation.

Of course, we want to display information about thousands of aircraft, not just one. For this, we can use a degenerate case of the GeoSPI display that renders only temporal information without any spatial representations (Figure 22). Now each aircraft becomes one line, which can be shown in an expanded view with the hourglass representations (shown on the top of the image) or in the compressed form (shown on the bottom of the image). In compressed form,

additional delays are shown in black and reductions in delays (catching up) are shown in white, with the aggregate delay shown in grey. We can see from this aggregate view that delays accumulate with time and become large in later flight segments. This concept offers developers the ability to see how flight delay propagates for a large number of aircraft, then switch to detailed views for a small number of selected aircraft, where the detailed views show the temporal and spatial aspects of the delay simultaneously in an easily understood format.

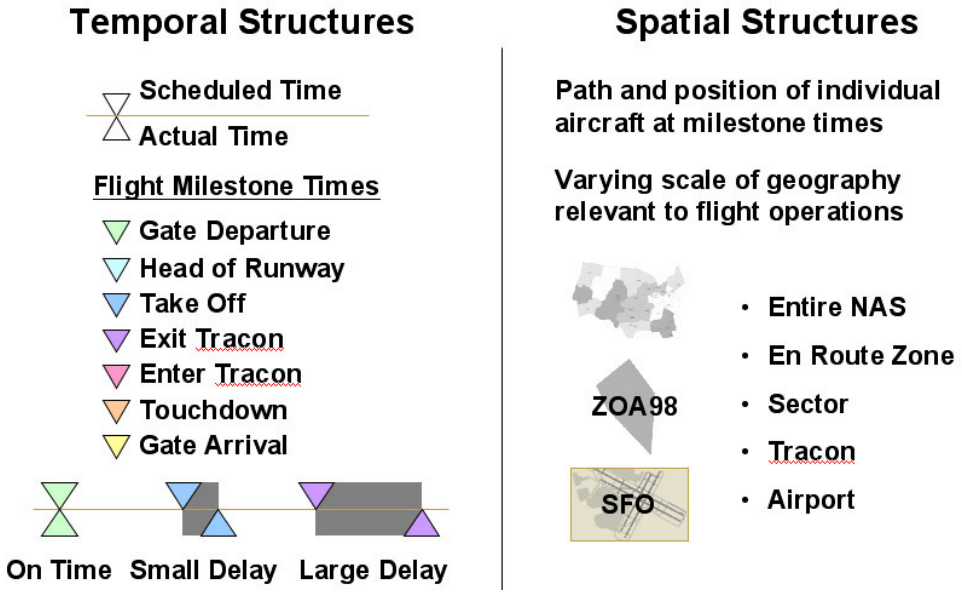


Figure 19. GeoSPI temporal and spatial structures.

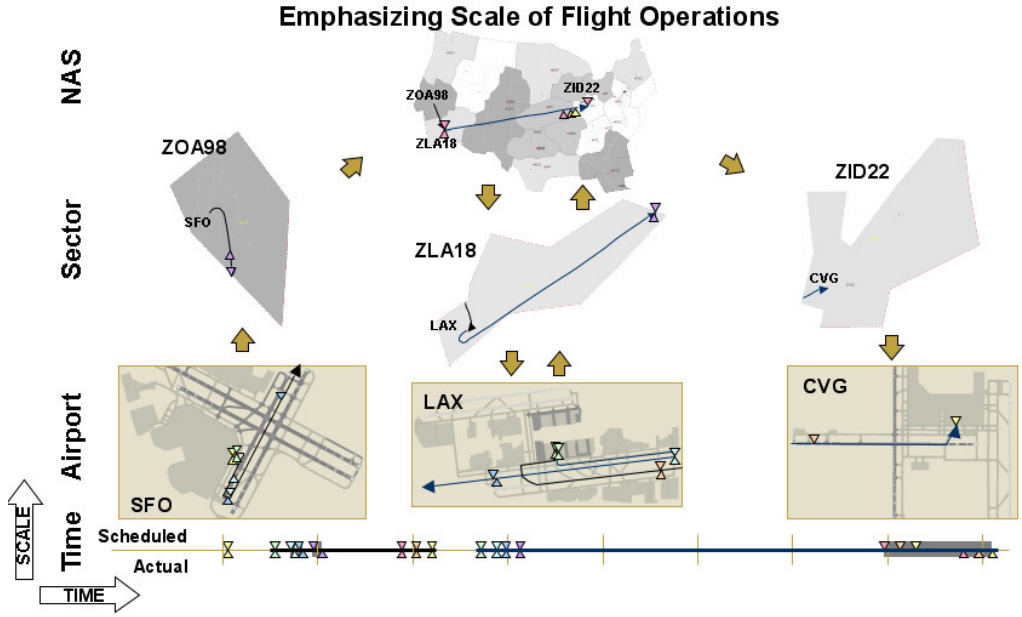


Figure 20. GeoSPI visualization concept, emphasizing scale of flight operations.

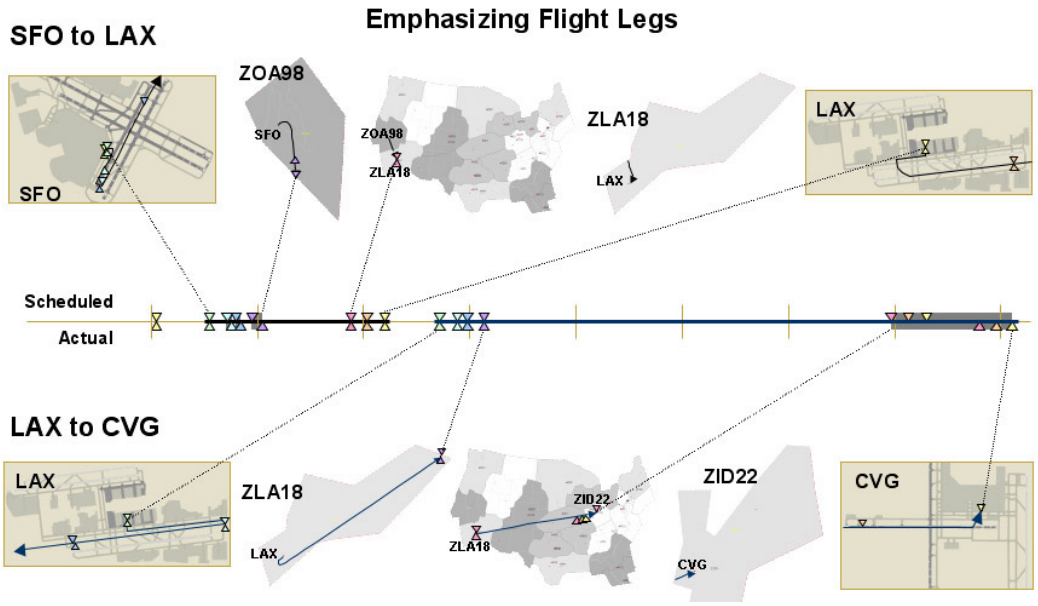


Figure 21. GeoSPI variation emphasizing flight segments.

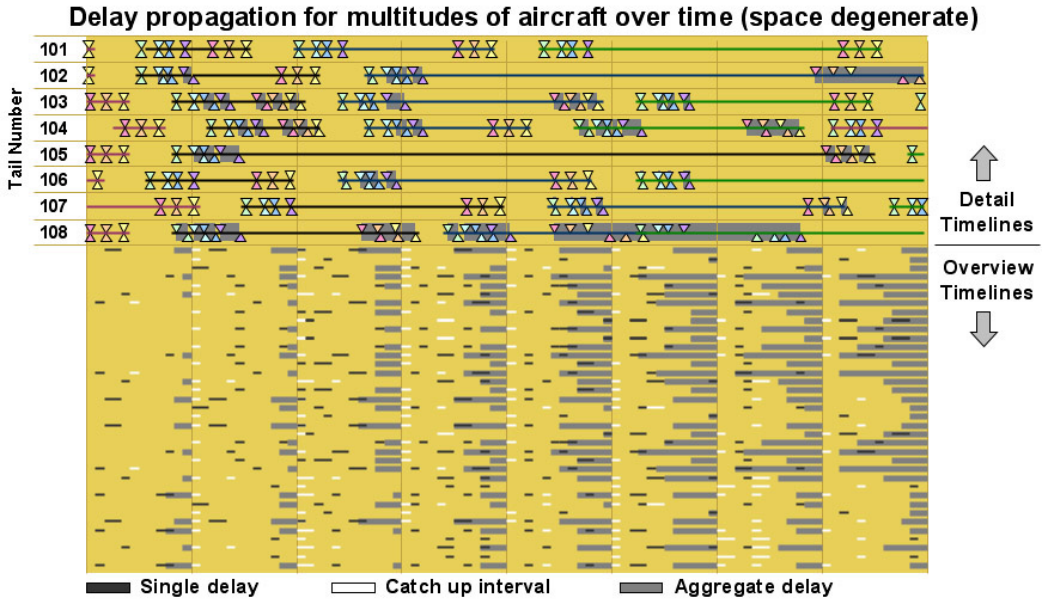


Figure 22. GeoSPI aggregate view (space degenerate).

4.9 Vector Field Alignment

The vector field alignment visualization concept is a self-organizing arrangement of aircraft glyphs that communicate delay problems by using human perceptual strengths in recognizing differences of orientation in adjacent lines.

Each aircraft is assigned a glyph that represents the delay for that aircraft. Figure 23 summarizes this glyph design. A line is used to indicate the aggregate delay. If the aircraft is exactly on time, the line is vertical. If the flight is cancelled, the line is horizontal (to indicate infinite delay). Varying amounts of delay are indicated by negatively sloped lines. A flight that is ahead of schedule has a positively sloped line. Bar graphs within the glyph indicate the components of delay or any other aircraft state.

Each square-shaped glyph must be assigned to one spot in a grid of squares. We want each glyph to be close to the true geographic location of the aircraft (as shown by drawing the grid over the map of the U.S.). To do this, we can use a self-organizing approach and define a cost function that penalizes moving aircraft glyphs away from their true location. An optimizing function such as simulated annealing can then search for the best assignment of glyphs to grid spaces that minimizes this cost function. In the actual implementation we did for this project, we used a much simpler greedy method that placed aircraft in order of their flight numbers. For each aircraft, it put the aircraft as close as it could to the actual spot where it was supposed to be. Since this is a greedy method, it will usually not yield optimal results, but this was easier to implement and faster to execute than a true optimizer (such as simulated annealing) would have been, and it proved sufficient to test the concept.

Figure 24 shows an example of what the end result could look like. This concept leverages human pattern recognition capabilities to see both similarities and differences in oriented lines. For example, an area of similarly sloped lines in the northeast section of the grid may indicate a common problem causing delays for aircraft in that region. However, seeing one aircraft that has radically different delay characteristics than its neighbors (such as the case on the bottom right) may indicate an outlier: something that affects only that aircraft (such as a mechanical problem) but does not affect the NAS as a whole.

We can also merge aspects of the vector field alignment concept with the GeoSPI concept. This hybrid shows a timeline for each aircraft. However, instead of using hourglass figures to indicate when certain milestones occur and how much delay is associated with each milestone, we render oriented lines for each milestone. If the line is vertical, then the scheduled time and actual time for that milestone are the same, and the aircraft is on schedule at that point. However, if the aircraft is delayed, that appears as negatively sloped vertical lines and highlighted grey regions. Since the milestones occur in the same order for each flight, the user can tell which milestone is which by counting the number that has occurred previously. Figure 25 summarizes this hybrid concept.

Figure 26 shows a detailed view from a preliminary implementation of the Vector Field Alignment concept on actual ACES data. The dataset represents twice the number of flights that actually occurred in the entire NAS for one particular high-traffic, low weather day. In this implementation, the direction the lines are rotated is reversed from the concept description. For example, delays are represented by lines that are rotated to the right (rather than to the left). In Figure 26, we can see an example of an outlier and also one cluster of flights with similar

amounts of delay. These are examples of effects that we thought might be revealed, as explained in Figure 23.

For a much more detailed discussion and evaluation of this concept, including more images of other datasets, please see the VisION evaluation report.

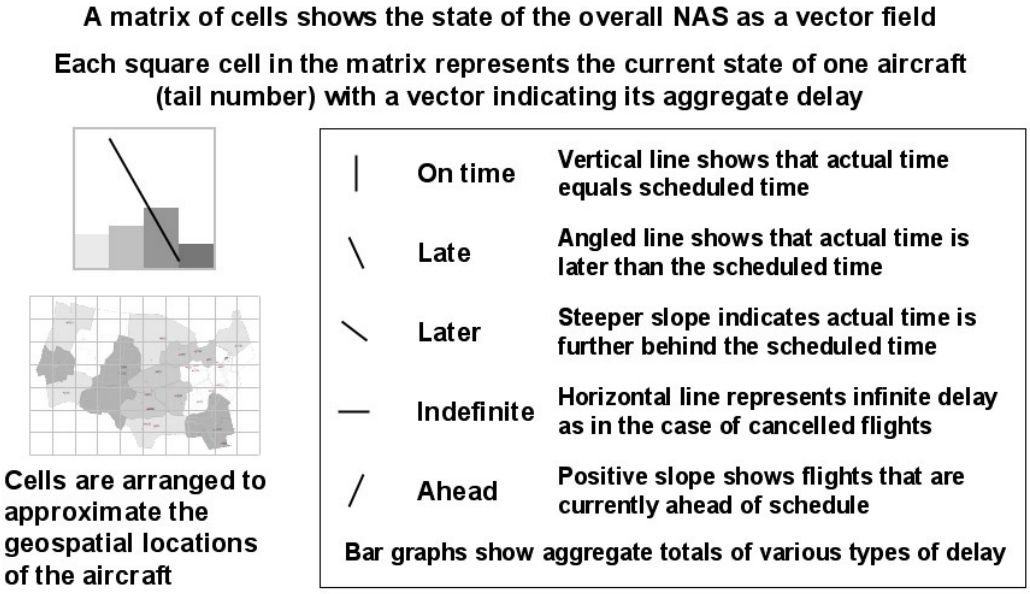


Figure 23. Vector field alignment glyphs.

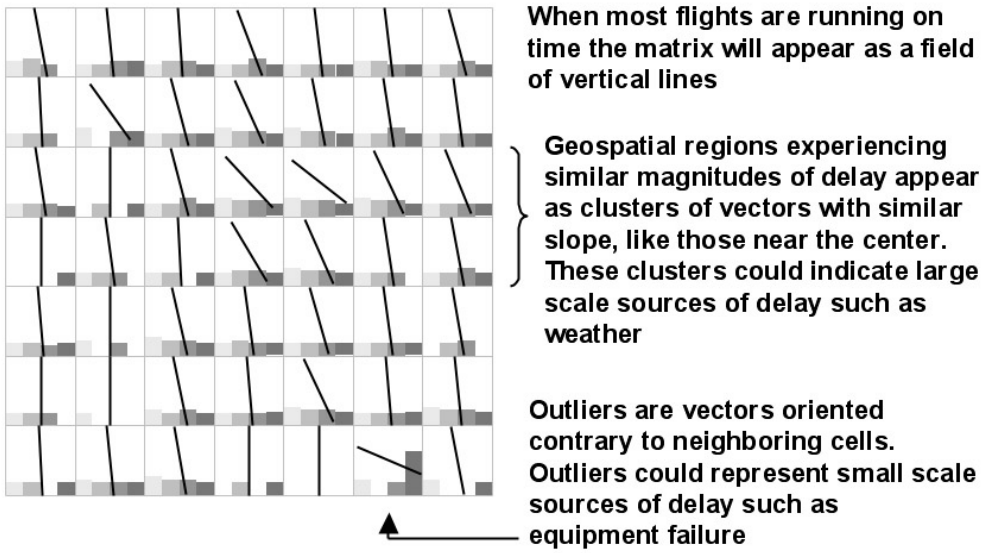


Figure 24. Vector field alignment visualization concept.

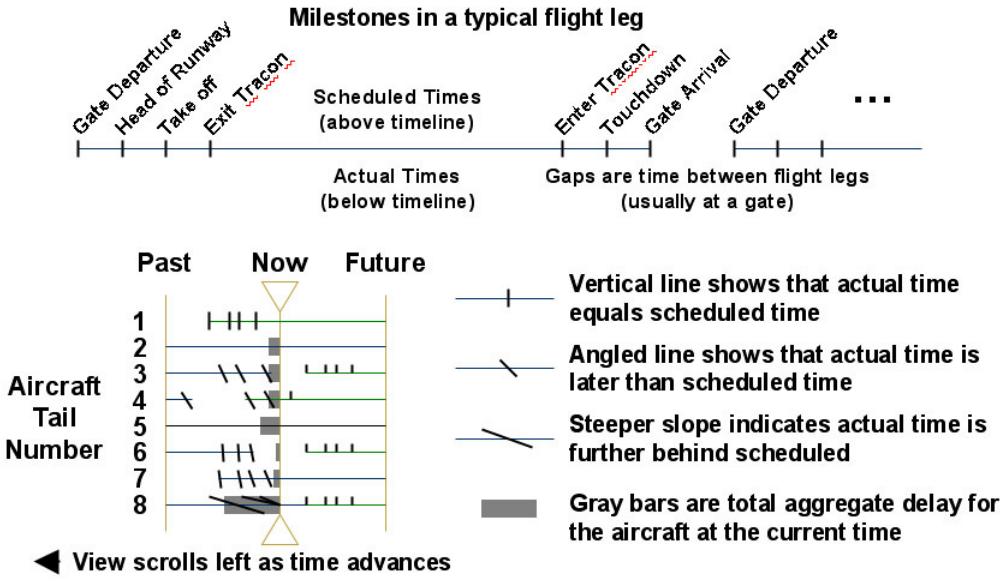


Figure 25. Timeline variation combining oriented lines with GeoSPI aggregate view.

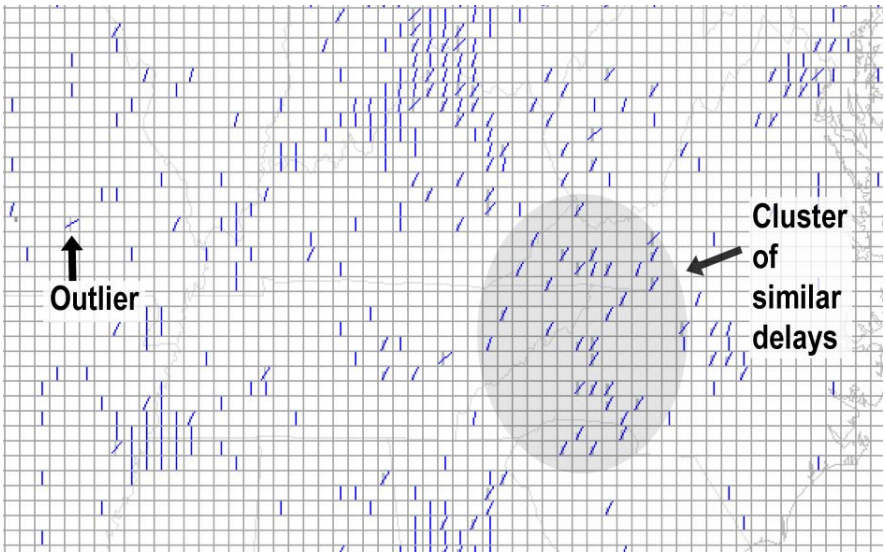


Figure 26. Example of detectable features in the Vector field alignment implementation. This is a closeup of an image of a 2×NAS dataset.

4.10 Flight Segment Aggregated Delay

ACES measures different components of delay. In the datasets that Raytheon provided, there were three types of delays: gate holds, taxiing out, and en route delays. However, a simulator could provide even more specific breakdowns of delay, including delays in the TRACON regions and delays at the arrival airport (in surface movement and delays going into the aircraft's designated gate).

The Flight segment aggregated delay visualization concept is intended to enable a user to interactively discover correlations between components of delay and any other NAS feature (such as airports, geographic regions) We developed this to address one of the user questions on how delay components are correlated with other aspects (such as geographical regions). This visualization concept focuses on flight segments. Each flight progresses in phases, from starting at a gate, to taxiing to the runway and departing, to leaving the departure airport's TRACON region, to the en route phase, and then to the arrival airport's TRACON, runway and taxiways, and finally docking at the arrival gate. We can represent each phase by an icon, where the icons are listed in temporal order, from left to right (Figure 27). We color each icon based upon the amount of delay associated with each phase.

Figure 27 shows an example for one individual flight. However, we can also use this representation for any aggregation of flights. This can apply to any possible aggregation of flights. Figure 28 gives an example of increasing aggregations. The bottom of the figure shows a small aggregation of flights (around SFO) while the top of the figure shows a larger aggregation of flights (all aircraft in the northwestern United States). The aggregated areas are indicated by icons of the U.S. on the Y axis, where the areas are the orange circles over the U.S. map. The user would select any type of aggregation of interest. This could be groups of airports, sectors, other geographic regions, airlines, types of aircraft, weather features, etc.

What types of correlations could this visualization concept reveal? Figure 29 shows one example: the aggregated delay of all flights affected by a particular weather feature. In this case, the weather problems prevent flights from departing, so the delays due to gate holds become very large. ACES does not directly model weather features, so we would not expect to actually see this in ACES data. However, this gives one example of a type of correlation that might be identified.

Another correlation involves showing both inbound and outbound flights for a particular set of airports (Figure 30). We can represent these by showing two groups of flights segments attached to each other, where the one on the left shows the aggregated inbound flights and the one on the right represents the aggregated outbound flights. In this case, the high amounts of delay in the outbound flights are correlated with inbound flight delays in the last segments. This shows a correlation where something that holds outbound traffic at their gates (due perhaps to weather) causes other delay problems for inbound flights, since the spots those inbound flights need to use are being occupied by the outbound flights that have not left.

Overall, this visualization concept is intended to allow a user to see how aggregated delays are distributed across flight segments. By interactively selecting different aggregations, a user might be able to find revealing correlations and insights between the aggregations and types of delays.

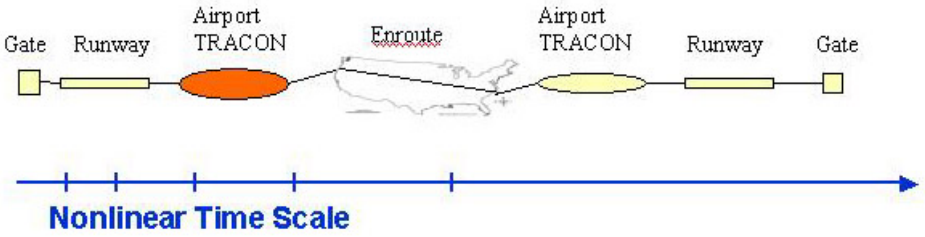


Figure 27. Flight segments.

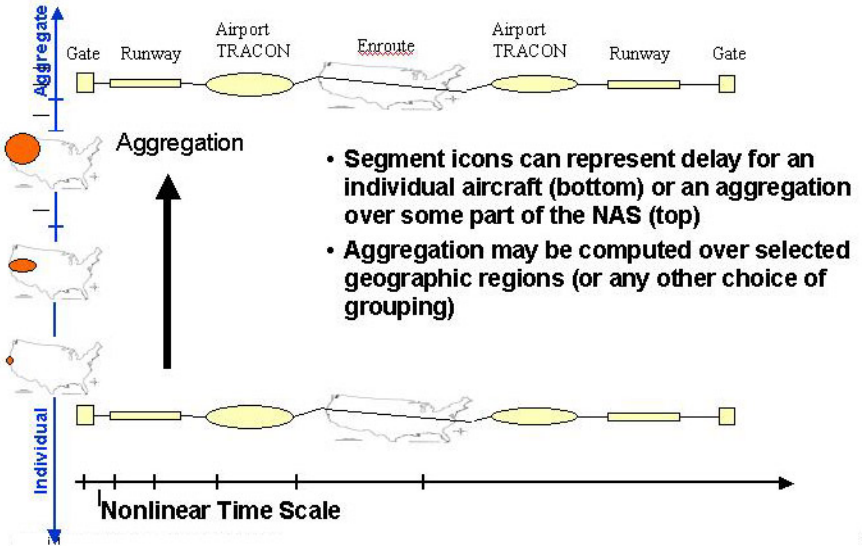


Figure 28. Aggregations of delay.

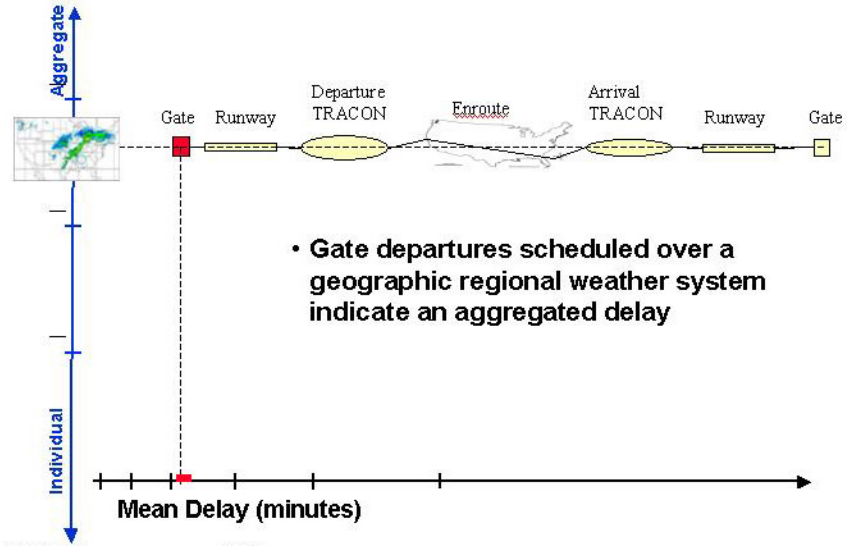


Figure 29. Aggregated delays due to a weather feature.

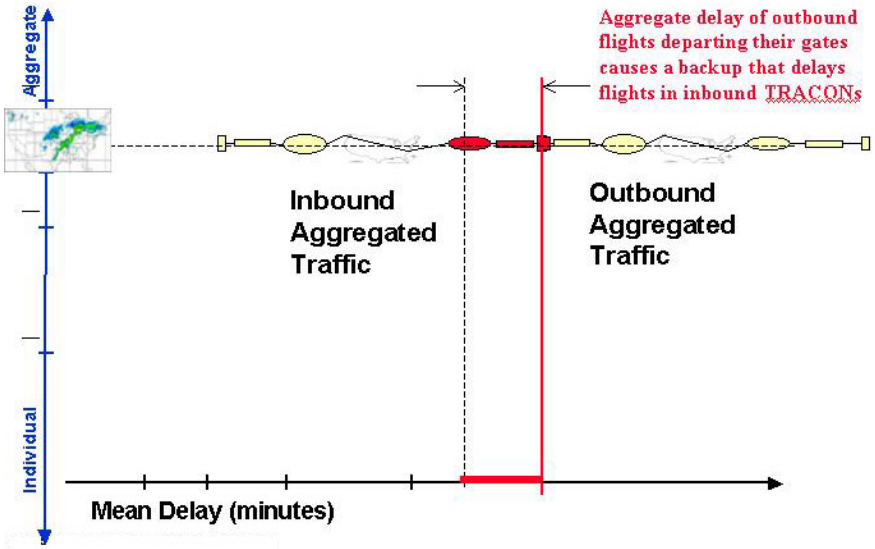


Figure 30. Aggregated delays for inbound and outbound traffic.

4.11 Self-similar Displays

The motivation to develop self-similar display modes comes from the belief that parts of the NAS might be similar to each other, at different times or at different places. Another motivation is the desire to compare related NAS measurements against each other. For example, traffic at one airport might be similar to another airport's. We also know the NAS exhibits cycles, such as the daily 24-hour cycle, or the groups of aircraft that arrive and then depart at a large airport in the hub and spoke system. The NAS also exhibits seasonal trends (e.g., demand is generally greater in summer than in winter). Furthermore, we believe VAMS concept developers would find it useful to compare the results of different capacity-enhancement concepts against each other, or to compare the same concept against different days in the NAS, or to compare the same concept with different settings against a baseline.

The information visualization community uses cyclic visualization techniques to enable such comparisons. For example, Matthew Ward and other researchers have introduced graphs that plot different settings along a spiral. Different glyphs along the same radial angle away from the center can represent the results at the same time of day, but for different days. This allows users to compare the glyphs at the same time of day across multiple days. There are other cyclical plots that similarly encode daily, weekly and monthly cycles to allow users to see temporal and seasonal patterns. However, these approaches generally plot very few items at each time. This would force us to encode the NAS state into just a few numbers, which runs the risk of hiding many interesting details. Furthermore, it is often difficult to compare items that are far away from each other in these schemes.

To apply some of these same principles but in a way that may be more appropriate for our application, we first encode the entire NAS state into a N by 1 vector (Figure 31, left). For example, this might represent the delay in all the sectors, where there are N sectors. Then we wish to compare the results of C concepts, across T timesteps. This results in a 3D array of data,

which naturally suggests a 3D plot. However, we need to consider how to plot this data. Many approaches would make it difficult to compare adjacent values along particular dimensions. As an example, see Figure 31, right. This suggests a "paddlewheel" metaphor, where wheel blocks of size N by C are arranged along the outside of a cylinder. As the wheel rotates, different timesteps T are brought to the top. This would make it easy to compare adjacent values in N and C, but very difficult to compare adjacent time values.

Therefore, we take the approach of plotting the data into a rectangular volume (Figure 32) of dimension N by T by C. This could be rendered as a single 3D entity using standard volume rendering techniques. The user could also examine cut planes through this volume. For example, planes that are perpendicular to the C axis would show the result of one concept at all time values. Cut planes that are perpendicular to the T axis would show the result of all the concepts at one point in time. These could be animated, or interactively flipped through. Such a concept might allow a user to examine all concepts, at all time values, for all N encoded states, in one object. Particular 3D features might correlate to particular problems in the NAS that the user might be able to quickly recognize, with experience.

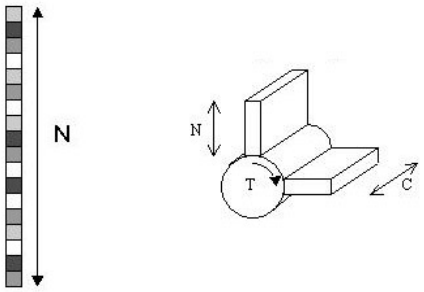


Figure 31. (Left) NAS state encoded into N by 1 vector. (Right) Example of 3D mapping that makes some comparisons difficult.

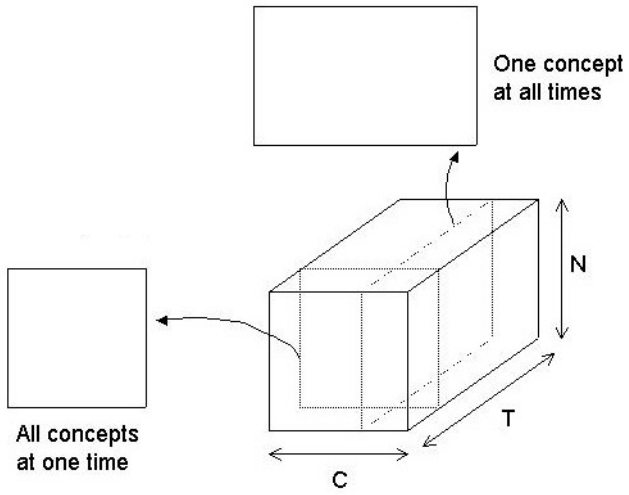


Figure 32. Self-similar visualization concept: volume slices.

4.12 Trend and Limit Display

David Payton of HRL helped contribute this concept. The goal of the Trend and Limit display is to make obvious which items in the NAS merit further examination, based upon how close their values are to some thresholds and the rate of change of those values. This can be used to address one of the concept developer questions of visualizing how resources (such as airports) are overloaded with time throughout the NAS.

First, we assign a 1D time series to each item of interest. For example, this might be the amount of congestion at each airport, or something that is derived from a combination of other values. In each 1D time series, we have *limits* and *trends*. Limits are boundaries of interest that are either hard numbers that should not be exceeded (e.g., capacity) or simply thresholds of interest (e.g., delay). Trends indicate the past history of the value. If the time series is "in trend," then the value has remained approximately constant in recent history. If the time series is "out of trend," that means the value is changing rapidly. Figure 33 shows an example of a time series with both a lower and upper limit. In the beginning, the value stays in trend but it is close to the lower limit. Near the end, the value goes out of trend but moves back toward the norm.

Once we have a large number of items, each with an associated time series, we can plot the location of each item inside a rectangle based upon how close the current value is to the limits of interest and whether the value is in trend or out of trend. Figure 34 shows where objects are plotted. Items that are near their normal value are plotted toward the top of the rectangle, while objects that are near a limit of interest are plotted near the bottom. Similarly, objects that are in trend are on the left side, while objects that are out of trend are placed on the right side.

By choosing to plot items in this manner, the user can quickly focus attention upon the items requiring the most attention. Figure 35 shows the objects that are in trend and have normal values (the upper left corner), are not a problem, and therefore do not require attention. Objects that are changing rapidly but are not near a threshold (upper right corner) may become a problem later but are not currently a problem, so they should be monitored. Similarly, objects that are near a threshold but not changing rapidly (lower left corner) can be a problem but not one that is progressing, so those deserve monitoring. The most attention should be reserved for the objects in the lower right corner. Those are the ones near a threshold and rapidly changing.

Figure 36 shows an example of what this visualization concept might look like with a set of airports. This shows that three airports (San Francisco, Seattle and Los Angeles) are rapidly moving toward the lower right corner, which is the area requiring the most attention. From that, the user might infer that there is a problem with major West Coast airports. By mapping NAS values into these 1D time series and plotting them in this fashion, this visualization concept will enable users to quickly recognize which items in the NAS require the most attention, simply by observing the locations of each object inside the rectangle and the patterns of how those objects move with time inside the rectangle. We can use this as a filtering mechanism that enables concept developers to focus their attention upon the key items and objects in the NAS that are overloaded or will become overloaded. Outliers and correlations will become obvious through this visualization concept.

Figure 37 shows an image from a preliminary implementation of the Trend and Limit concept on a 2×NAS ACES dataset. In this implementation, the Y axis has been flipped so that the green quadrant (normal and in trend) is the lower left rather than the upper left. In this sample image, we have plotted both en route zones and sectors with respect to the total delay of the aircraft inside that zone or sector. The "limit" is defined as the largest total delay that occurs within any zone or sector during the entire simulation. We use a nonlinear mapping, so that only the sectors and zones that are getting close to the limit appear in the upper half of the image. The data blocks move around significantly with time. This noise is due to the inherent nature of these measurements, since an aircraft that transitions from one sector to another immediately removes its delay from the original sector and adds it to the new sector. Also, the rate of change is computed essentially by numeric differentiation, which is a noisy process. Nonetheless, this image does show that the concept identifies a handful of sectors and zones that are approaching the limit, or are rapidly changing, or both, and therefore might be worth further investigation by users examining this dataset. For example, Figure 37 shows that the sector "ZDV21" is changing rapidly and has a large amount of delay.

The VisION evaluation report provides more images of test datasets and evaluations of our preliminary implementation.

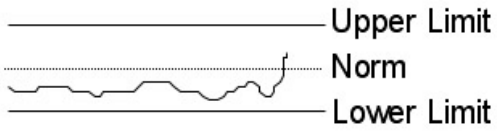


Figure 33. Example of a 1D time series.

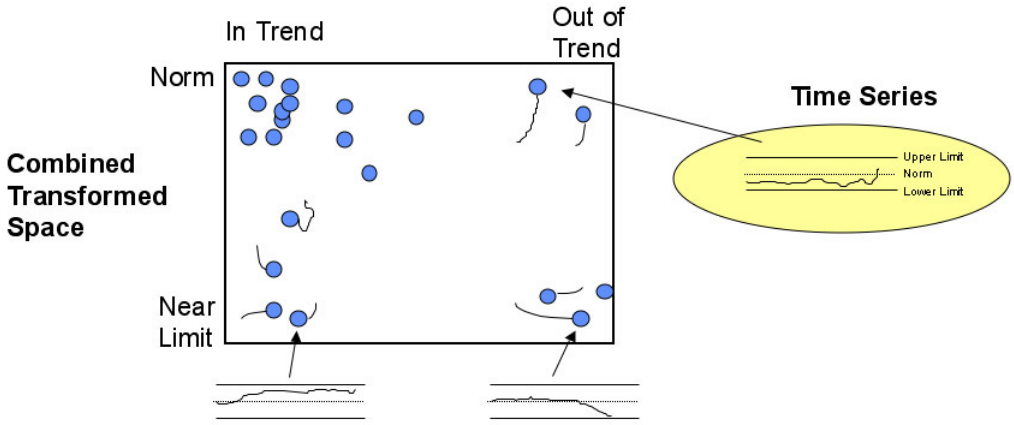


Figure 34. Trend and Limit plot.

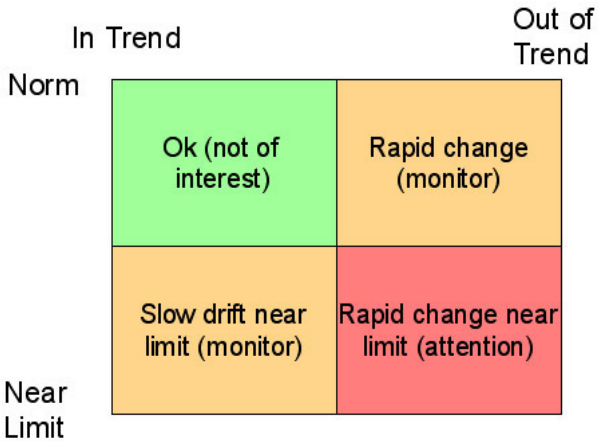


Figure 35. Areas of user attention in the Trend and Limit visualization.

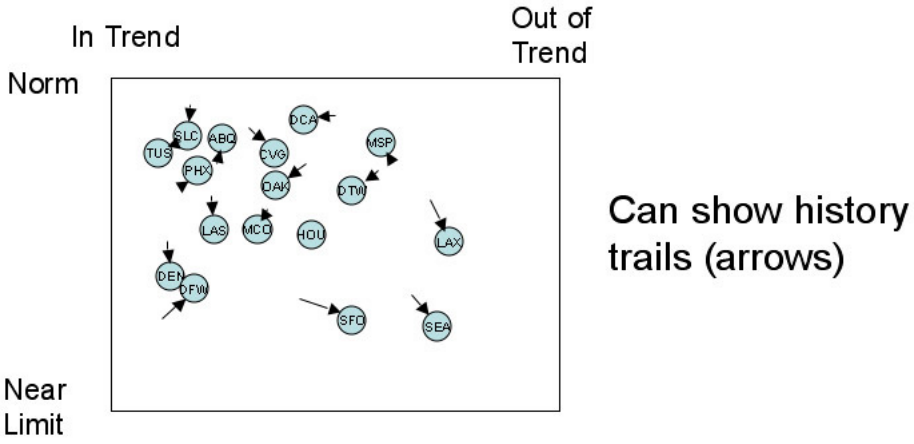


Figure 36. Trend and Limit visualization concept example.

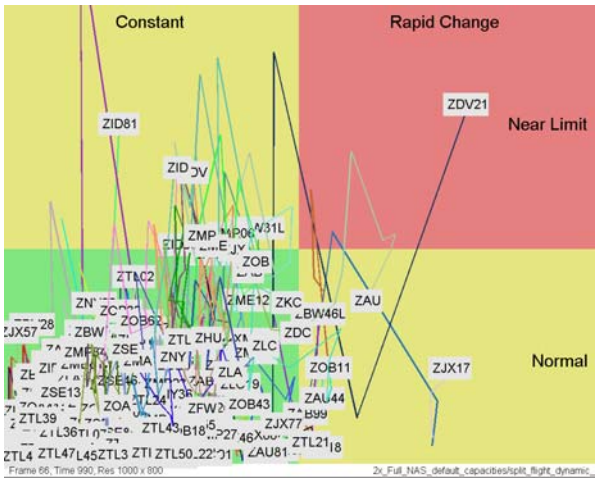


Figure 37. Implementation of Trend and Limit concept on a 2×NAS dataset.

5.0 ACES DATASETS

The subcontract to Raytheon called for Raytheon to provide at least one ACES dataset that we could use in this project, along with a description of the format. Raytheon also led the effort to ask the VAMS concept developers what types of questions they wanted a visualization to answer. We documented the results of that in the mid-project review and in Section 3.2. Raytheon was also required to host one meeting to demonstrate ACES and aid the transfer of the data. This occurred in Marlborough, MA on March 23, 2005.

Raytheon provided several ACES datasets. We included five on the DVD-R disc with the software and evaluation report. Three of those are 2×NAS datasets, which have double the normal amount of traffic for a particular day (Feb. 19, 2004) that was chosen because it was a high traffic day with few weather problems. The other two are 2×ORD datasets, which filters the traffic to only include flights that either depart from or arrive at Chicago O'Hare (ORD). The 2×ORD datasets were primarily useful for debugging and testing purposes, since they are not realistic. One of the 2×NAS datasets and one of the 2×ORD datasets were not useful for our evaluation because they assumed essentially infinite sector and airport capacities. Therefore there were no significant congestion or delay problems, and nothing to visualize.

The specifics of each dataset, along with a detailed description of the format of the data, are listed in the software description document (VisION_software_description.doc). Raytheon is also providing the scripts used to generate the ACES data files and a brief description of how to use those scripts. Those are in a supplemental document included on the CD-ROM with this final report.

6.0 SOFTWARE IMPLEMENTATION SUMMARY

We implemented the key visualization cues for three of our concepts. These were:

- Airport – Airspace Correlation
- Trend and Limit Display
- Vector Field Alignment

These were written in Matlab for quick prototyping.

The details of the code, the data, their locations, and how to execute and modify these programs, are all covered in the software description report in VisION_software_description.doc.

7.0 EVALUATION SUMMARY

We evaluated the implemented concepts by running the concepts upon the ACES datasets we had available and recording our observations. In some cases we found unexpected features, which we noted. We also offered general guidelines and rules of thumb based upon the lessons we learned.

Because the evaluation required many pages, we put it into its own document, the Vision evaluation report: VisION_evaluation_report.doc. Please refer to that for the details of the evaluation.

The evaluation, due to the constraints of this project and the limited amount of data we had, is by necessity preliminary in nature.

8.0 CONCLUSION AND FUTURE WORK

We successfully completed all required tasks in this contract, and demonstrated that our visualization concepts have potential to aid in the task of understanding NAS simulation data. Since the main focus was in the generation of ideas, the implementation and evaluation are preliminary, but promising.

In the VisION evaluation report we documented direct evidence that the visualizations we designed did trigger the perceptual cues we expected, when tested on the datasets we had available. Also, we were able to observe and record specific examples of unexpected traits that would be difficult to otherwise find, or for the user to know to ask that specific question. Therefore, we think these preliminary evaluation results are promising. However, to generate a more compelling case requires a follow-on effort. This effort would focus on generating interactive, real-time versions of these visualizations that are integrated with other data exploration tools, to enable the user to make use of our concepts as a filter that identifies a small subset of the dataset for further investigation. Furthermore, we would use these integrated tools upon ACES data that came from multiple simulation runs from the VAMS concept developers' capacity-enhancement concepts. Ideally, these tools would even be delivered to the concept developers themselves. Testimonials of the usefulness of these visualization tools, when given by the actual users performing the intended application, are strong evidence of the value of the visualization concepts. We hope to have the opportunity to continue this work along this direction.

Appendix A – Scripts

1.0 At the VisION final presentation at AFRL Rome, Jason Moore asked us to include the scripts that Raytheon used to generate the ACES datasets we included on the DVD-R on Sept. 21, 2005. Greg Trott and Mary Ellen Miller, both of Raytheon, have provided those scripts and the description that we list in this document.

The two Perl script files are:

```
sector_count.pl
flight_dynamic_data_splitter.pl
```

These are both included on the CD-ROM along with this report and the VisION final report.

2.0 DESCRIPTION OF SCRIPTS

2.1 General Notes

The provided Perl scripts extract ACES run data from a MySQL server and process it to produce the output files.

Before use, 4 variables must be set in each script:

$db_name - the MySQL database of interest.

$db_host - the hostname of the MySQL server.

$db_user - the user name with which to log on to the MySQL server.

$db_password - the password for $db_user.

2.2 sector_count.pl Script

To run this script, bring up a command window and enter the following command to run the sector_count.pl script. This script generates counts of the number of aircraft in each sector and TRACON in 15 minute slices:

```
perl sector_count.pl
```

This script produces two output files:

```
sector_count.csv
tracon_count.csv
```

These files provide comma separated values of both types of counts.

2.3 flight_dynamic_data_splitter.pl

The source code is written in Matlab, and the source code files are .m files. We used two versions of Matlab running on Windows XP service pack 2 PCs, and verified that the code works on both versions. The version numbers are 7.0.4.365 (R14) service pack 2, and 7.0.1.24704 (R14) service pack 1. The Airport-Airspace and Trend and Limit concepts use only basic Matlab functions.

To run this script, bring up a command window and enter the following command to execute the flight_dynamic_data_splitter.pl script.

```
perl flight_dynamic_data_splitter.pl [milliseconds between reports]
```

This script reports the status of each flight at specific moments in time. Status information includes:

- Whether the flight is scheduled for departure
- Whether the flight has arrived at the destination gate
- Latitude, longitude, and sector name for airborne flights
- Gate departure delay (after the flight has left the gate)
- Taxi out delay (after the flight has taken off)
- In-flight delay (after the flight has landed)
- Taxi in delay (after the flight has arrived at the gate)

If [milliseconds between reports] is not specified, it defaults to 3600000 = 1 hour. For the datasets that Raytheon provided to HRL, 900000 milliseconds were specified to produce reports at 15 minutes intervals.

This script produces multiple output files with names like "timestep_000.csv". Each output file contains a timestamp followed by the status information for each flight for that timestamp.