

# Data visualization techniques for airspace flow modeling

Aude Marzuoli, Christophe Hurter, Eric Féron

► **To cite this version:**

Aude Marzuoli, Christophe Hurter, Eric Féron. Data visualization techniques for airspace flow modeling. CIDU 2012, Conference on Intelligent Data Understanding, Oct 2012, Boulder, United States. pp 79 - 86, 2012, <10.1109/CIDU.2012.6382187>. <hal-01022432>

**HAL Id: hal-01022432**

**<https://hal-enac.archives-ouvertes.fr/hal-01022432>**

Submitted on 21 Jul 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Data Visualization Techniques for Airspace Flow Modeling

Aude Marzuoli  
*Georgia Institute of Technology*  
Atlanta, GA, USA  
aude.marzuoli@gatech.edu

Christophe Hurter  
*ENAC*  
Toulouse, France  
christophe.hurter@enac.fr

Eric Feron  
*Georgia Institute of Technology*  
Atlanta, GA, USA  
feron@gatech.edu

**Abstract**—With the predicted growth of air traffic, traffic flow managers need new tools to access information to support their decision making processes. Recent progress with information visualization tools enables users to explore large data sets and extract decisive knowledge. Their advantages for air traffic applications are presented in this paper. They can provide high level information to aggregate trajectories. With constant feedback due to human perception, a flow model of the airspace, reflecting its intrinsic structure, is elaborated and can be used for further research.

## I. INTRODUCTION

The goal of Traffic Flow Management is to maximize both safety and capacity, so as to accept all flights while protecting the lives of the passengers and avoiding delays. With the expected growth of air traffic, new visualizations and analysis tools need to be developed to help the engineer design a more efficient and safer airspace/air traffic control system/airtraffic management system and further improve the safety level. In this perspective, air traffic practitioners analyze data from traffic activity. Such multidimensional data includes aircraft trajectories (3D location plus time), flight routes (ordered sequences of spatio-temporal points that represent planned routes), meteorological data, etc.

In many Traffic Flow Management tools, the initial position of the aircraft (known with or without uncertainty) is the key information. Mid-term and long-term en-route air traffic management would benefit from a higher view approach, using tracks and flows characteristics (e.g. routes, flight plans, inter-arrival distances between aircraft). Prandini et al. claim that [1] flows can provide more predictable and robust estimates than watching aircraft individually. As developed in [2], [3], dominant flow patterns and traffic flow interactions constitute the basis for air traffic controllers to build their abstraction of a sector. Flow patterns can be described by many flow features [4], such as the number of flows, the major flows and their size, the number of crossing flows, etc. These indicators are useful to estimate and predict sector demand based on the traffic flow pattern. They can be further used to study the impact of severe weather, as explained in [5]. The list of flow characteristics proposed in [4] can also form a basis for the notion of traffic complexity. The locations of high complexity regions is of special interest for traffic flow managers [6].

Traffic models are needed to best represent the airspace and the variability and complexity of traffic. In many research approaches for TFM [7], a graph network is formed linking chosen airports. Often times, the network includes the entries and exits of each sector along the way. The resulting network is used to formulate TFM optimization problem. Yet, modeling traffic as if aircraft were simply traveling along a limited set of predefined air routes and jet routes from an origin to a destination is unrealistic. Obtaining a more precise understanding of the airspace and its capacity is a means of improving the support for TFM under nominal conditions, but also in the presence of perturbations. Data-mining techniques are useful to identify and interpret the behavior of a system. Extracting knowledge from large data sets with no prior knowledge enables researchers to build better predictive models. The specific properties of air traffic data offer both new challenges and new opportunities for data analysis. The semantics of traffic data is rich because of its temporal and spatial ranges. More semantics can be added by augmenting background data such as the traffic network and the meteorological data. Air traffic datasets are characterized by their large sizes, adding more challenges to the analysis.

This paper seeks to present new data visualization techniques and their purposeful use for building airspace flow models. Section 2 focuses on the challenges of data visualization and its application to large trajectory data sets. Section 3 develops the use of graph bundling techniques to build network flow models of an airspace. Section 4 draws the conclusions of this paper and suggests future research paths.

## II. TRAJECTORY EXPLORATION AND ITS APPLICATION TO LARGE DATA SETS

Much of air traffic data consists of sets of trajectories, which often contain inaccuracies or errors. This section first presents the direct manipulation requirements for trajectory exploration, then the way users can interact with large sets of trajectories and finally two applications in air traffic.

### A. Direct manipulation of trajectories

An aircraft trajectory is a record of positions of an aircraft in a given airspace (3D positioning with time and other

information such as identifier, speed etc). Exploration of trajectory with direct manipulation techniques helps to:

- Understand past conflicts and then improve safety with adequate evolutions,
- Assess new onboard and ground safety systems and the resulting aircraft trails,
- Devise new air space organization and procedures to handle traffic increase,
- Compare trails with environmental considerations (fuel consumption, noise pollution, vertical profile comparison),
- Study profitability from a business trajectory point of view (number of aircraft on a specific Flight Route per day, number of aircraft that actually landed at a specific airport),
- Filter and extract trajectories in order to re-use them (this task will be later illustrated in this paper in the section on trajectory extraction for Air Traffic Controllers training).

FromDaDy [1] (which stands for FROM DATA to Display) was developed in this sense. It is a visualization tool that tackles the challenge of representing, and interacting with numerous trajectories (several millions trajectories composed of up to 10 million points). FromDaDy employs a simple paradigm to explore multidimensional data based on scatterplots, brushing, pick and drop, juxtaposed views and rapid visual configuration. Together with a finely tuned mix between design customization and simple interaction, users can filter, remove and add trajectories in an incremental manner until they extract a set of relevant data, thus formulating complex queries.

### B. Views Organization and Navigation

A FromDaDy session starts with a view displaying all the data in one scatterplot. The visualization employs a default visual configuration, e.g. the mapping between data dimensions and visual variables. The view is inside a window, and occupies a cell in a virtual infinite grid that extends from the four sides of the cell. The user can configure the two axes of each scatterplot and use other visual variables such as color and line width to display dataset dimensions. For instance, in Figure 1, the dataset field latitude is attached to the Y axis, and the field longitude to the X axis. The altitude is used to color trajectory sections, so that low altitudes are in green and high altitudes are in blue.

### C. Trajectories Manipulation

A simple and efficient direct manipulation technique is implemented: trajectories brush, pick and drop. The user selects a subset of the dataset by means of a brushing technique. Brushing is an interaction that allows the user to brush graphical entities, using a size-configurable or shape-configurable area controlled by the mouse pointer. Each trajectory touched by this area is selected, and becomes gray.

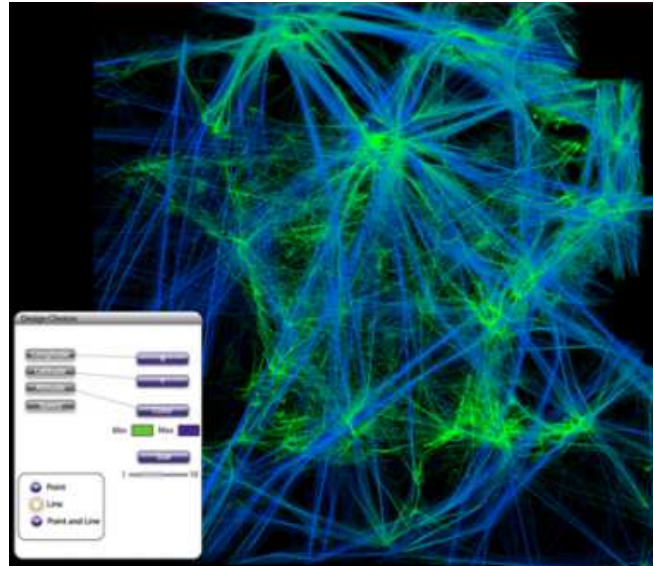


Figure 1. One day record of traffic over France. The color gradient from green to blue represents the ascending altitude of aircraft (green being the lowest and blue the highest altitude). The French coastline is apparent here in terms of pleasure flights by light aircraft and the straight blue lines represent high altitude flight routes. A user interface shows the dataset fields and the defined visual configuration.

The selection can be modified by further brush strokes (Ctrl key pressed), or by removing parts of it with brush strokes in the erase mode (Shift key pressed). The display shows a brush trail, so that the user can see and remember more easily how the selection was made. The combination of fast switching between the add/erase mode, trajectory visualization, rapid size-setting, and cursor-centered zooming allows for fast and incremental selection.

Then the user can pick brushed trajectories by hitting the space bar. Previously selected data can be extracted from the current scatterplot and attaches it to the mouse pointer so it appears in a fly-over view (transparent background). When the user hits the space bar for the second time, a drop occurs in the view under the cursor. If the view under the mouse pointer is empty, the software creates a new scatterplot with the selected data. If the space bar is pressed while moving over a non-empty view, FromDaDy adds the selected data to this scatterplot. Although it resembles a regular drag and drop operation, we prefer to use the term pick and drop, because the data is removed from the previous view and is attached to the cursor even if the space bar is released. A view can be removed if the brush selects all the trajectories and if the user picks them.

### D. Brush Pick and Drop Assets

The fundamentally new aspect of FromDaDy compared to existing visualization systems, is to enable users to spread data across views. Within FromDaDy, there is a single line per trajectory instance: trajectories are not duplicated, but

spread across views. The advantage of this technique is multifold. It enables the user to remove data from a view (and drop it on to the destination view). The fly-over view enables the user to rapidly decide if the revealed data (previously hidden by the picked data) is interesting. Secondly, it makes it possible to build a data subset incrementally. In this case, the user can immediately assess the quality of the selection, by seeing it in the fly-over view. Furthermore, by removing data from the first view, the user makes it less cluttered, and this makes it easier for him to pick and drop more trajectories.

Another asset of the Bush pick and Drop paradigm is that this interaction helps the user to perform complex Boolean operations such as the selection of trajectories that belong to a given area but not the ones that are too high and only those that are faster than a given minimum speed. A seminal previous work uses containers (also called layers) to cluster trajectories and explicitly applies Boolean operations to combine them. Even with an astute interface, Boolean operations are cumbersome to produce, since results are difficult to foresee. FromDaDy overcomes this drawback, since all the operations of the interaction paradigm (brushing, picking and dropping) implicitly perform Boolean operations. Removing trajectories corresponds to an XOR operation and dropping trajectories corresponds to an ADD operation. The following examples illustrate the union (AND), intersection (OR) and negation (NOT) Boolean operations. With these three basic operations the user can perform all kinds of Boolean operation: AND, OR, NOT, XOR (Figure 2, Figure 3).

In Figure 2, users want to select trajectories that pass through region A or through region B. They just have to brush the two desired regions and Pick/Drop the selected tracks into a new view. The resulting view contains their query, and the previous view contains the negation of the query. Figure 3, the same process is used to find the trajectories that pass through A and B. By sequencing two pick and drop operations, users formulate their request.

#### *E. Example of application: Outliers identification*

In this scenario, recordings of aircraft trajectories over France over one day is exploited. In this dataset, a unique and incremental identifier is assigned to each trajectory. The first trajectory's index is 0, the next one's is 1, etc. Figure 3 shows an abstract visualization of this dataset. The X axis on the screen shows the time of each radar plot and the Y axis on the screen shows the aircraft identifier. Since these identifiers are incremental over the day, the resulting visualization shows a noticeably continuous shape, in which each horizontal line represents the duration of one flight. The slope of the shape indicates the accumulation of traffic during the day (due to the incrementally assigned identifiers). Hence, the traffic notably increases at 5 am and decreases at 10 pm as reflected in the change of slope. The

width of this shape indicates the average flight duration in the dataset: it is about 2.5 hours, which represents the average time taken by aircraft to cross France. But some aircraft have longer trajectory durations. The user brushes these long trails (the ones that come out of the curved shape). When visualizing them with a latitude (Y screen) and longitude (X screen) visual configuration, the user discovers a figure of eight shaped trajectory. This trajectory covers 6 hours and performs 11 loops. After further investigation, it was found that the trajectory corresponds to a military tanker plane.

The user would have also been able to perform the same data extraction task with a textual tool, like SQL queries. The only difference is that, a textual tool would not have led the user to the idea of exploring long flight duration in order to extract military aircraft. Only with the incremental trajectory exploration, can the user discover the valid requests for this dataset. In a sense, the user explores the dataset, and at the same time, explores the request to perform. Even if this process is efficient, the direct manipulation cannot be automatic. Analysts need interactive tools to enhance their exploration capabilities. The power of interactive graphical tools is now discussed in depth for an airspace analysis task.

#### *F. Example of application: defining an airspace structure*

The data used to build an airspace model is taken from the Enhanced Traffic Management System (ETMS) data set for the United States. The Cleveland center (ZOB ARTCC) is selected because a large share of traffic to and from the Northeast of the United States flies through it, and it is known to be one of the most congested centers in the United States. The ETMS data corresponds to aircraft radar tracks (longitude, latitude, altitude), collected during 123 days, between May and August 2005, this data was previously filtered for inconsistencies. 270,007 trajectories remain and are resampled to eight points each. This corresponds to more than two millions points to explore with FromDaDy.

This large data set can be easily observed and manipulated with FromDaDy. Using an accumulation map on the x and y coordinates, and adapting the transparency of the trajectories represented, the most dense areas in the airspace become clearly visible. Such areas correspond to "highways of the sky", as shown in Figure 5. The intersections of these routes engender areas of high complexity associated with heavy traffic loads which require careful monitoring and significant effort by the controller to manage. Such a representation is similar to the presence maps described in [6], that indicate the local density of traffic. These maps can provide traffic flow managers with information regarding the complexity and difficulty required to manage a given airspace.

Also, en-route traffic and ascending or descending traffic can be separated for further specific analysis. This is done by using a vertical view in FromDaDy, brushing trajectories with a portion under FL 100, picking them and dropping

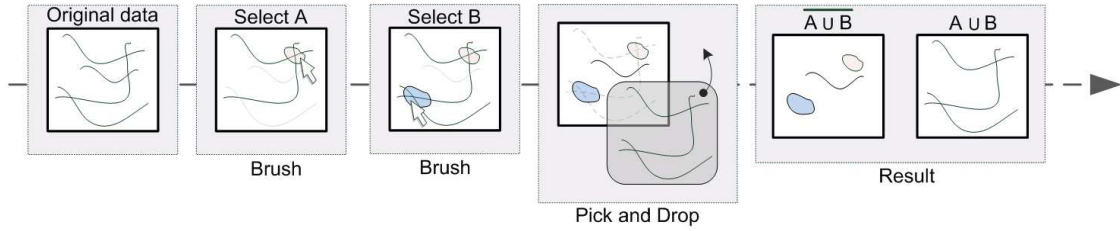


Figure 2. Union Boolean operation.

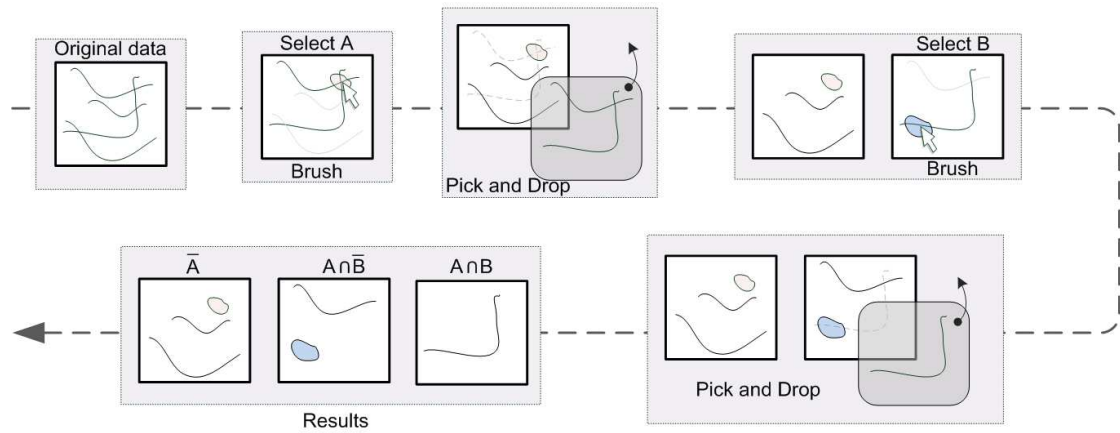


Figure 3. Intersection Boolean operation.

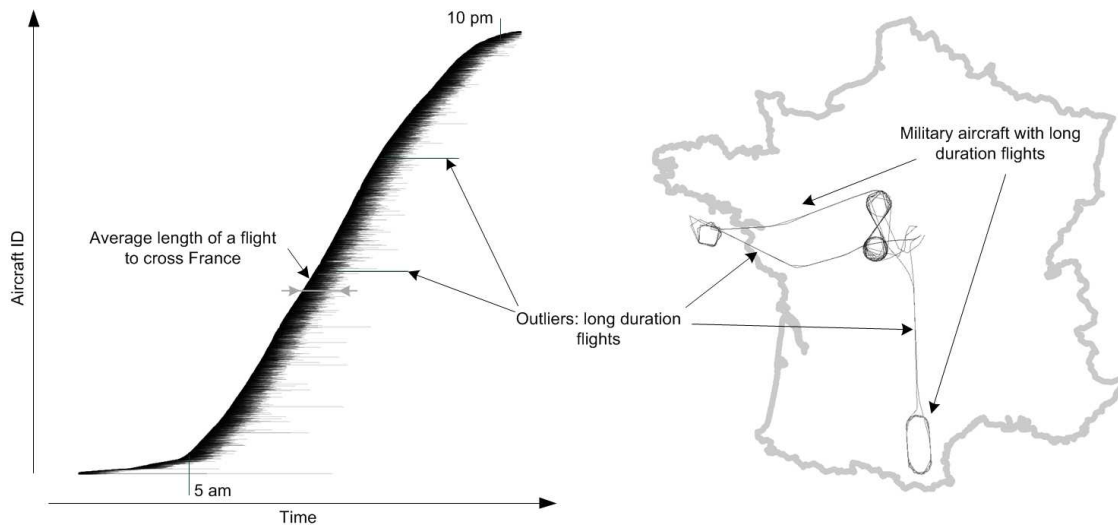


Figure 4. Detection of supply planes with an abstract visualization

them into another window. In the  $xy$  plane, as shown in 6, the ascending and descending traffic makes the airports in the Cleveland center clearly identifiable (CLE, DTW and PIT for instance). It further highlights the variability of trajectories due to a large range in aircraft performance and pilot decisions.

### III. AIRSPACE FLOW MODELING THROUGH GRAPH BUNDLING

#### A. Kernel Density Estimation-based Edge Bundling (KDEEB)

In recent years, graph bundling methods have gained increased attention. Bundling starts with a set of node positions, given as input data or computed by a layout



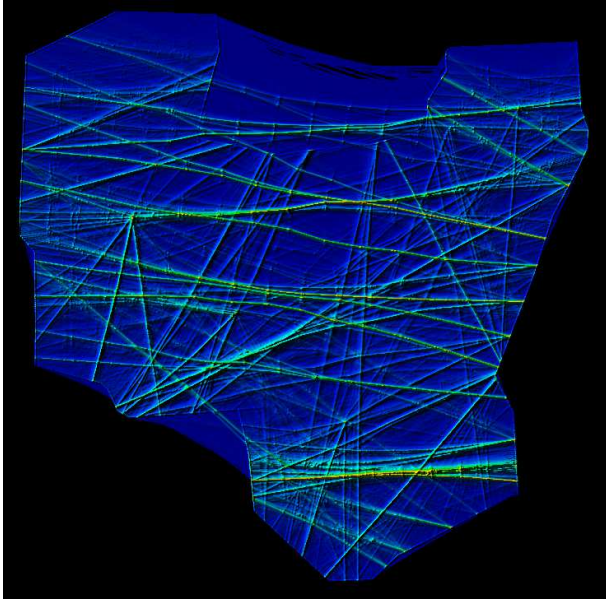


Figure 5. Accumulation Map of three months of traffic in the Cleveland center.

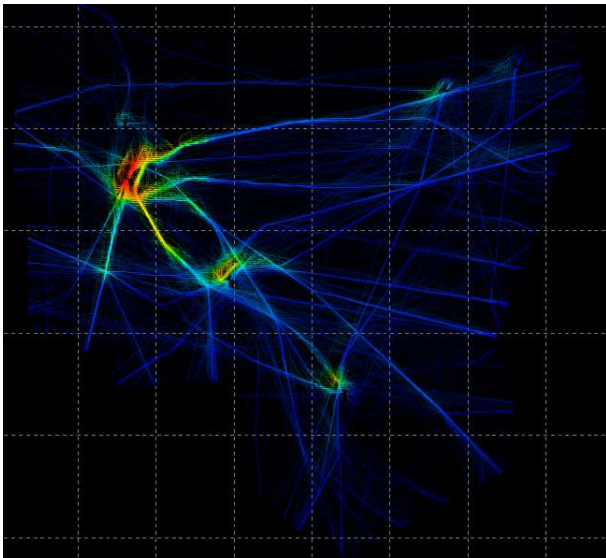


Figure 6. Ascending and descending traffic in the Cleveland center.

algorithm. Edges which are close in terms of graph structure, position, data attributes, or combinations thereof, are drawn as tightly bundled curves. This trades clutter for overdraw and produces images which are easier to understand and/or better emphasize the graph structure.

A new method for bundling general graphs was presented in [10]. The algorithm behind it is entirely graphical. Kernel density estimation is a non-parametric way to estimate the probability density function of a random variable. Kernel density estimation is a fundamental data smoothing problem, where inferences about the population are made, based on a

finite data sample. For the present case, given a graph drawing, the existing edges are re convolved with a special kernel to construct a density map. Next, it advects edges in the gradient of this map and iterates the process for a few steps with decreasing kernels size. This delivers a layout with well separated and smooth bundle structures. This bundling technique is applicable to general graphs and it is robust, simple to implement and up to one order of magnitude faster than other state-of-the-art bundling techniques. Some of its challenges are to intuitively control the look and feel of the bundling (e.g. produce smooth or ramified bundles) and to provide an easy implementation (no complex parameter settings or algorithms).

A fast and simple method is chosen to compute bundled layouts of general graphs. A given graph drawing is transformed into a density map using kernel density estimation. Then an image sharpening technique is applied, which progressively merges local height maxima by moving the convolved graph edges into the height gradient flow. Moreover, the edges are smoothed to remove discontinuities introduced by the gradient computation. Finally, the edges are re sampled to maintain a uniform sampling rate for each of the edges. Figure 7 details the computation pipeline to produce a bundled graph. Figure 8 shows, an illustration of the bundling process with detailed integration steps. Several iterations of the density map are detailed, drawn as a height plot (normalized in height for display) and corresponding bundled layouts for the US migration graph. The density map gets sharper during the iterative solving. This bundles edges along the local maxima density. As the density map gets sharper, the average distance between local maxima increases, so the bundles get tighter and separated by more empty space.

### B. An airspace flow structure

In the previous section, FromDady was used to better visualize the main routes in the airspace with ETMS data. However, these routes are closely entangled, and isolating them to identify the flows is, at first, complicated, over such a large data set. The airspace may be represented as a flow network, a set of routes (edges) with intersections (nodes) that enable aircraft to travel to their destinations. KDEEB can be used to declutter and separate the flows from each other, as well as localize the main "nodes" or areas with high probability of conflict. This is done in two steps:

- Use KDEEB to bundle the trajectories into a less cluttered graph,
- Use FromDaDy to extract each flow, composed of bundled (i.e. distorted) trajectories.

One of the main issues is to control the outcome of the bundling. The resulting graph needs to be sufficiently decluttered to enable the separation of the flows, and yet not lose the structure and complexity level of the airspace. The final graph should not be too distorted, the airspace

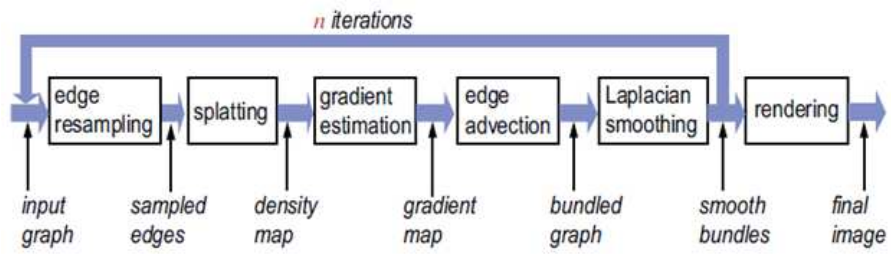


Figure 7. Pipeline to obtain a bundled graph.

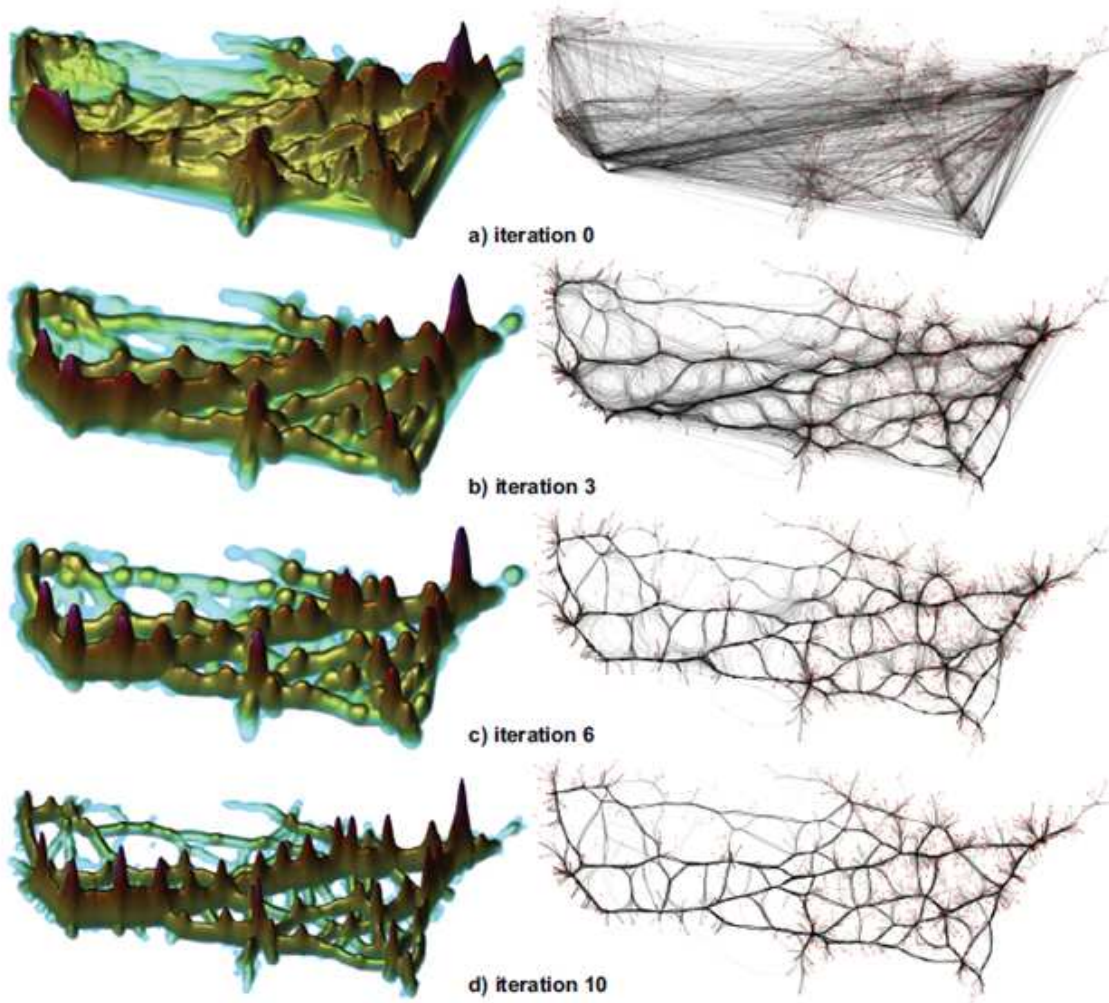


Figure 8. Iterative results of the bundling process.

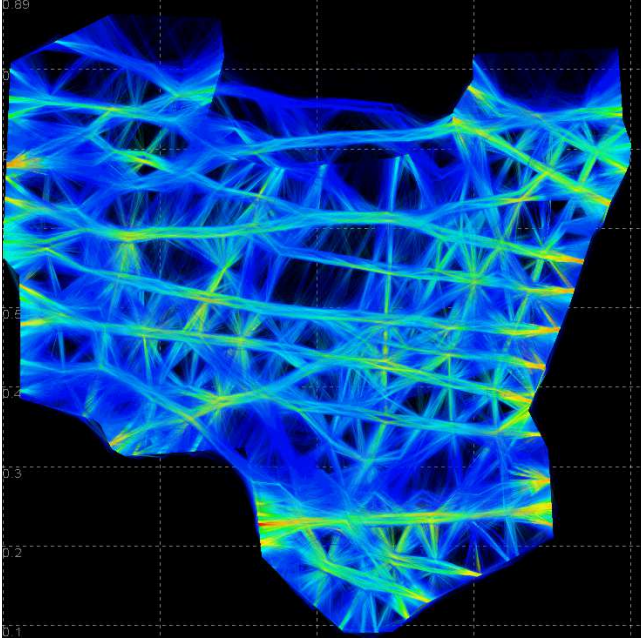


Figure 9. Graph bundle obtained from ETMS data.

structure apparent in the accumulation maps needs to be preserved. Three main parameters are used: the kernel size, the accumulation map size and the smooth factor. Approximately 15 iterations are needed to obtain a final result. One of the prominent advantages of this technique is the feedback obtained at each iteration through the visualization to finely tune the result. Figure 9 shows the graph obtained from KDEEB, visualized in FromDaDy with an accumulation map. The main flows are now sufficiently far apart for separation and further processing. The number of flows can be adjusted with the bundling, depending on the precision of the outcome wanted.

To model the airspace, the flows are collected with FromDaDy, through a succession of brushing, picking, dropping and naming each flow. This process leads to the identification of a total of 105 flows in the airspace. Figure 10 shows one of the flows extracted, and it can be noted that some trajectories have been slightly distorted. The original trajectories, that served as input for the bundle, can be retrieved with the trajectory identifier. The entry and exit points of the airspace are identifiable at the boundaries of the center. The "nodes" of the airspace can be located at the intersections of the flows. Around 40 entry and exit nodes and approximately 80 intersection nodes are identified. FromDaDy also enables the user to store the coordinates of the nodes in a click. To draw a comparison, the same procedure was applied to only one day of traffic (August 3rd 2005), that is 2,649 trajectories. It led to the extraction of 37 flows, 16 entry and exit points and 21 intersection points. Of course, the flows contained considerably less trajectories than when dealing

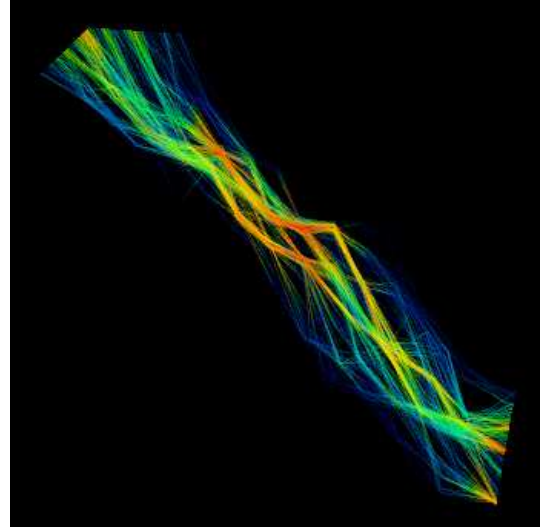


Figure 10. Example of a flow extracted in FromDaDy after the bundling.

with three months of data. It also highlights the fact that this procedure leads to an extensive model of the airspace, that is not entirely occupied on any day. It should be noted that, to the best of our knowledge, this is one of the first applications of graph bundling techniques.

### C. Graph representation of the airspace

The result obtained by bundling ETMS data is a visual graph, with a multitude of edges corresponding to the trajectories. This graph is still very complex, with the same number of edges. But the network we seek to extract is now clear to the eye, since the trajectories have been aggregated into clusters, or flows. The methodology for building such a graph has been presented in [11]. It is based on a flow representation, similar to the one extracted in the previous subsection. The edges represent portions of flow corridors in which aircraft fly, whereas the nodes correspond to where aircraft may enter, change, or leave a flow corridor. To generate the network, the following steps were carried out. First, the regions where flows spatially interact are located. These areas include intersections of flows and flow merging, and engender a high probability of conflict. Then, the edges that link the nodes of the network are computed, to recreate the possible flow routes an aircraft can travel on. On each flow, an edge is defined between all consecutive nodes (whether entry, intersection, exit) along the flow.

The strength of the data visualization approach compared to the one presented in [11] is that the dimensionality of the problem has been further reduced. In the cited paper, 690 flows obtained with the same data set lead to an airspace flow graph of 1288 nodes and 3085 edges. A more compact network is obtained here with the combined use of FromDaDy and KDEEB. With 105 flows, and a total of 120 nodes, 210 edges are defined for the en route



traffic flow model. First, this corroborates the approximate estimate of the Cleveland center complexity obtained when interviewing former air traffic controllers of this airspace. Second, in [11], the network flow model of the airspace is used for optimizing traffic in the airspace using estimates of controller taskload. If the network representation of the airspace obtained through data visualization is more compact, i.e. with less nodes and edges, the formulation of the traffic optimization can be extended to the entire United States, without computational issues. Furthermore, the work presented here was only focused on en route traffic for the sake of simplicity. Yet, the airports can be linked to the en route network with their own smaller networks, joining the main traffic at specific intersection nodes in the airspace. The traffic optimization formulation could then incorporate airports and delays.

Such a graph representation would be an entirely data-based support for the implementation of advanced integer optimization of large-scale air traffic flow management, such as the one presented by Bertsimas, Lulli and Odoni [12]. This network should also be studied from a graph theory perspective, to determine its connectivity and resilience to perturbations. The purpose is to identify the possible weaknesses in the airspace structure.

#### IV. CONCLUSION

This paper focused on the use of information visualization techniques and their specific use for air traffic analysis and model building. Two specific tools were presented, FromDaDy to explore large multidimensional data sets in an intuitive fashion, and KDEEB, to bundle huge graphs. By providing a graphical visualization of databases and queries, users can find trends and exceptions easily. Several applications were developed. The identification of outliers, such as military aircraft, was detailed. The complete extraction of the structure of traffic in the Cleveland ARTCC was explained, from extracting en route traffic with FromDady, to bundling the trajectories to diminish the clutter, to extracting the flows and nodes in FromDady. The methodology to build a network model from the flows were recalled from [11], that can serve as a basis for further traffic optimization.

This research is in continuing progress. Our next research steps are to build a network flow model with this process of a much larger airspace, such as the continental United States. The network model could then be used to optimize traffic, using estimates of controller taskload based on flows, and incorporating airport capacities, and delays.

#### REFERENCES

- [1] M. Prandini, V. Putta, and J. Hu, "A probabilistic measure of air traffic complexity in 3-d airspace," *International Journal of Adaptive Control and Signal Processing*, 2010.
- [2] J. Histon, R. Hansman, G. Aigoïn, D. Delahaye, and S. Puechmorel, "Introducing structural considerations into complexity metrics," 2002.
- [3] J. Histon, R. Hansman, B. Gottlieb, H. Kleinwaks, S. Yenson, D. Delahaye, and S. Puechmorel, "Structural considerations and cognitive complexity in air traffic control," in *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, vol. 1. IEEE, 2002, pp. 1C2–1.
- [4] L. Song, D. Greenbaum, and C. Wanke, "Predicting sector capacity for traffic flow management decision support," in *6th AIAA Aviation Technology, Integration and Operations Conference*, 2006.
- [5] —, "The impact of severe weather on sector capacity," in *8th USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)*, Napa, California, USA, 2009.
- [6] E. Salaün, M. Gariel, A. Vela, E. Feron, and J.-P. Clarke, "Airspace statistical proximity maps based on data-driven flow modeling," *AIAA Journal of Guidance, Control and Dynamics*, 2011.
- [7] D. Bertsimas and S. Patterson, "The air traffic flow management problem with enroute capacities," *Operations Research*, pp. 406–422, 1998.
- [8] B. Shneiderman, "Direct manipulation: A step beyond programming languages (excerpt)," *Readings in Human-Computer Interaction: A Multidisciplinary Approach*, pp. 461–467, 1987.
- [9] C. Hurter, B. Tissoires, and S. Conversy, "Fromdady: Spreading aircraft trajectories across views to support iterative queries," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1017–1024, 2009.
- [10] C. Hurter, A. Telea, and O. Ersoy, "Graph bundling by kernel density estimation," in *Computer Graphics Forum journal (EuroVis 2012)*, 2012.
- [11] A. Marzuoli, M. Gariel, A. Vela, and E. Feron, "Air traffic optimization on data-driven network flow model," in *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*. IEEE, 2011, pp. 3C1–1.
- [12] D. Bertsimas, G. Lulli, and A. Odoni, "An integer optimization approach to large-scale air traffic flow management," *Operations Research-Baltimore*, vol. 59, no. 1, p. 211, 2011.