Real-Time Animated Visualization of Massive Air-Traffic Trajectories

Draft Version

Stefan Buschmann, Matthias Trapp, Jürgen Döllner
Computer Graphics Systems
Hasso Plattner Institute
Potsdam, Germany
Email: {stefan.buschmann, matthias.trapp, juergen.doellner}@hpi.uni-potsdam.de

***Abstract*—With increasing numbers of flights world-wide and a continuing rise in airport traffic, air-traffic management is faced with a number of challenges. These include monitoring, reporting, planning, and problem analysis of past and current air traffic, e.g., to identify hotspots, minimize delays, or to optimize sector assignments to air-traffic controllers. Interactive and dynamic 3D visualization and visual analysis of massive aircraft trajectories, i.e., analytical reasoning enabled by interactive cyber worlds, can be used to approach these challenges. To facilitate this kind of analysis, especially in the context of real-time data, interactive tools for filtering, mapping, and rendering are required. In particular, the mapping process should be configurable at run-time and support both static mappings and animations to allow users to effectively explore and realize movement dynamics. However, with growing data size and complexity, these stages of the visualization pipeline require computational efficient implementations to be capable of processing within real-time constraints. This paper presents an approach for real-time animated visualization of massive air-traffic data, that implements all stages of the visualization pipeline based on GPU techniques for efficient processing. It enables (1) interactive spatio-temporal filtering, (2) generic mapping of trajectory attributes to geometric representations and appearances, as well as (3) real-time rendering within 3D virtual environments, such as virtual 3D airport and city models. Based on this pipeline, different visualization metaphors (e.g., temporal focus+context, density maps, and overview+detail visualization) are implemented and discussed. The presented concepts and implementation can be generally used as visual analytics and data mining techniques in cyber worlds, e.g., to visualize movement data, geo-referenced networks, or other spatio-temporal data.**

***Keywords*-spatio-temporal visualization, trajectory visualization, 3D visualization, visual analytics, real-time rendering**

I. INTRODUCTION

In spatio-temporal data analysis and visualization, research focuses on the development of methods that facilitate the human perception and understanding of temporal aspects, e.g., concurrency and temporal order of events [1]. In addition to static depictions, animation and interactive exploration based on cyber worlds provide effective means for presenting, analysing, and understanding of temporal aspects.

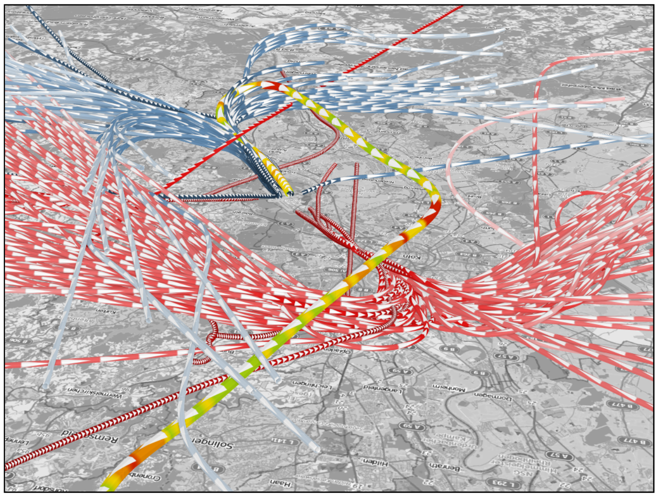


Figure 1: Exemplary visualization of aircraft trajectories using color mapping, texturing, and animation. The image shows departing (blue) and arriving (red) aircrafts. For a single highlighted aircraft, animated arrows are used to depict direction, while acceleration is visualized by color.

In recent years, visual analytics has emerged as a research field in the context of spatio-temporal visualization and analysis [2], enabling analytical reasoning with interactive systems that combine methods of scientific visualization, data mining, and interaction.

The analysis and visualization of movement represents one fundamental category within this field. In particular, the visualization of movement trajectories, often embedded in a geographic context, represents a key functionality. This includes the representation of spatial and temporal aspects of movement data as well as depicting additional data attributes of complex attributed trajectories. Challenges include: (1) to cope with the complexity of the data, (2) to detect and communicate spatial and temporal phenomena, and (3) to handle the typically large data sets, while maintaining flexibility and interactivity for, e.g., the analyst user.

In this paper, we propose a real-time animated visualization technique for massive complex movement trajectories embedded in a 3D virtual environment (Fig. 1). It supports 3D trajectory visualization, including a configurable mapping to visualize static attributes and dynamics, and aggregation techniques such as the generation of 2D density maps. To achieve visualization and exploration at interactive frame rates, all stages of the visualization pipeline are implemented using GPU techniques, including filtering, mapping, and rendering. This enables interactive spatio-temporal filtering, configurable mappings of complex attributes to visual properties, and real-time rendering of large data sets. Based on this approach, several visualization metaphors, such as temporal focus+context [3], density based analysis, and space-time cube visualizations, are implemented and evaluated on the use case of air-traffic data. To summarize, this paper makes the following contributions:

1) Description of a fully GPU-based visualization pipeline, including filtering, dynamic mapping, and real-time rendering of complex trajectory data.
2) Application of the described visualization pipeline to implement different visualization techniques for the interactive visualization of movement data.

The remainder of this paper is structured as follows. Section II briefly describes the application context and the exemplary data set used in this work. Section III reviews related work in the fields of visual analytics of movement data and interactive visualization of 3D trajectories. Section IV describes the concept and implementation of our visualization pipeline. Section V demonstrates application examples of the presented approach. Finally, Section VI outlines conclusions and future research directions.

II. PROBLEM STATEMENT

This paper presents an approach for interactive visualization of 3D air-traffic trajectories, aimed at developing decision-support systems in the context of air traffic management (ATM). In this context, processing and visualization of large sets of 3D trajectories at interactive frame rates, as well as providing tools for interactive analysis and exploration to analyst users, are the main challenges. In the following section, the application context, use cases, and the resulting challenges are described in more detail.

A. Application Context

The field of air traffic management (ATM) contains operational tasks, i.e., air traffic control (ATC), as well as a broader spectrum of management tasks, in particular, airspace, flow, and capacity management. While ATM is mainly concerned with monitoring and managing the current situation in air traffic, it also includes the analysis of the past and the planning for the future, i.e., by modifying flight routes or adjusting regulations. In this context, the analysis of air plane movements can be an important tool for analysing and understanding past events, for assessing current events, and finally for drawing conclusions for the future. An objective may be to analyze past events and derive insights from it, as to understand the cause of events and learn how to prevent them or optimize processes. Exemplary questions and interests for the analysis of air plane movements include:

1) Flight path analysis: By investigating the distribution of movements over time and space, areas with higher and lower traffic volumes can be identified. From this, common flight routes, which are used often, can be derived. Trajectories can be categorized according to those flight routes.

2) Classification: A comparison of flight routes allows for identifying similarities and differences between different flights. By correlating them to other flight attributes (e.g., departing and arriving flights, or different aircraft types), a classification of flights can be achieved.

3) Pattern discovery: A further goal is to identify typical and untypical movement behaviors (i.e., detect outliers) and if possible correlate them to spatial or temporal circumstances. In this context, categorization and clustering methods [4], [5] can be employed to categorize trajectories.

4) Adherence to safety regulations: By using additional data such as flight routes, air traffic sectors, or safety regulations, flight traffic that violates those can be identified. A goal is to find reasons or circumstances of such violations and if possible to compare consequences of administrative or regulatory actions regarding to air safety violations.

5) Comparison of alternative routing scenarios: For air traffic planning, one requirement is to compare alternative scenarios. For example, the results of changing flight routes, or the consequences of how a new air port is designed, can be examined by simulating and comparing the resulting movement trajectories.

B. Data Set and Data Acquisition

The test data set used in this paper comprises true 3D trajectories of aircraft movements and is a typical example for aircraft movement data. It originates from RADAR tracking around an airport and includes departing and arriving IFR (instrument flight rules) flights. The tracking has a temporal resolution of four seconds and a range of about 50-70 km. Therefore, each single trajectory has a duration of 5-10 min with approximately 75-150 sample points. For a duration of one month, the data set consists of a total number of 12,500 trajectories and more than 1,500,000 sample points.

In addition to spatial and temporal information, the flight data contains attributes per-trajectory and per-point. Per-trajectory attributes include a flight identification (ICAO callsign), the type of aircraft, and meta data about the flight from the airport (i.e., arrival/departure, time stamps, and runway designations). Per-point attributes include a time stamp, the current geographical location, as well as the current height and velocity over ground.

C. Visualization of Trajectories

To support the analysis of the described data set with respect to the tasks and questions mentioned above, interactive visualization and visual analytics tools based on cyber worlds can be applied. For the data set at hand, this requires tools that support real-time visualization of large numbers of 3D trajectories, their spatial and temporal features, as well as their complex attributes. In addition, interactive exploration in both, the spatial and the temporal domain, needs to be supported. This enables users to explore a data set, identify spatial and temporal phenoma, correlate them to trajectory attributes and thereby gain insight into the data.

Visualization methods have to be designed to work on large amounts of data, to find patterns and phenomena in past events. For a single international airport, this amounts to approximately 10,000 individual trajectories per month. Extending the scope of analysis either spatially or temporally, e.g., by applying it to real-time data, quickly results in a massive amount of data that has to be processed and visualized. Another challenge is the demand for interactivity, to provide a visualization system in which users can explore data interactively. Finally, the inherent 3D nature of the data set poses additional challenges to the visualization system.

D. Challenges in 3D Air-Traffic Visualization

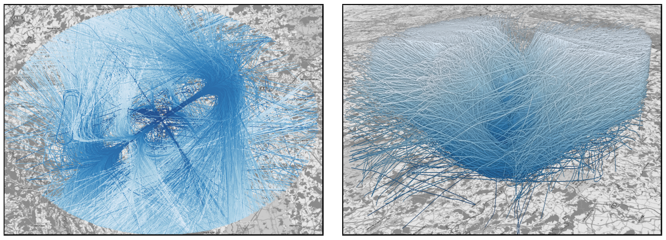
In the discipline of spatio-temporal analytics, in particular the analysis of moving objects, visualization and analysis is mainly based on 2D representations of movements. For example, pedestrian movements in a city can be expressed and visualized using 2D information, while the 3D-position is often not required and therefore omitted.

In our use case, though, movements need to be visualized in 3D space, since their movement characteristics significantly depend on all three dimensions. For example, focussing on aircraft trajectories, the detection of flight routes, common flight corridors and spatial and temporal hotspots demand for a true 3D analysis. Therefore, the actual 3D positions of aircrafts have to be maintained and expressed in a visualization. However, temporal aspects have to be conveyed as well. This poses a number of challenges for 3D visualization techniques of spatio-temporal data:

1) Visualization of temporal aspects: Since all three axes are used to display the spatial components of the data itself, no axis is left to express the temporal information. Therefore, temporal information has to be mapped to other visual properties, such as color, width, or texture, or has to be visualized in other forms, e.g., by means of animation.

2) Perspective foreshortening: When using perspective projections of 3D scenes onto planar surfaces, spatial positions appear perspectively distorted on the image. Also, perception of height and size can be difficult and is prone to misinterpretation. This effect is inevitable, in particular when using non-stereo displays, but must be recognized, since it can disturb the cognition of the displayed data.

3) Occlusion and clutter: Occlusion poses another major problem for 3D visualization of massive air-traffic data [6]. Trajectories overlap each other, thereby occluding objects behind them. A large number of objects displayed at the same time also leads to visual clutter (Fig. 2). These problems can be approached by filtering (omitting the rendering of objects that are currently not of importance), selection (highlighting important objects), and transparency (making objects semi-transparent, to retain the overall context by partially visualizing occluded objects).



(a) 2D Movement Trajectories. (b) 3D Movement Trajectories.

Figure 2: Comparison between 2D and 3D visualization of a massive set of movement trajectories.

III. RELATED WORK

Visualization approaches for spatio-temporal data are often either inherently two-dimensional or use the third dimension to display information other than the spatial 3D component of the input data (e.g., time or other attributes of the data).

Space-time Cube: The space-time cube [7] uses a three-dimensional representation of space and time: X- and Y-axes are used for spatial components, while time is mapped to the Z-axis. Therefore, it is especially suitable for visualizing two-dimensional data with a temporal component, but may not be easily applicable to visualize true 3D data. In recent years, the space-time cube has become a popular visualization method in the field of geovisualization and visual analytics [8], and several frameworks based on the space-time cube have been developed [9], [10]. Kristensson et al. have conducted a user study on the space-time cube [11], identifying trade-offs regarding the use of a space-time cube for presenting spatio-temporal data to users. Their results indicate that the space-time cube can support the comprehension of complex spatio-temporal data. Kveladze et al. have developed a methodological framework for evaluating the usability of the space-time cube [12].

Density Maps: Density maps can be used to aggregate movements and to represent the influence of movement trajectories on their neighborhood using a regular grid. Willems et al. used kernel density estimation to aggregate vessel movements in a harbor [13], applying different kernel sizes to distinguish between historical and current movements, which enables users to identify current movements

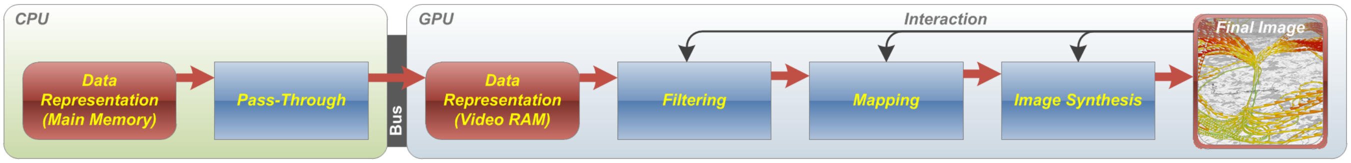


Figure 3: Architecture of the GPU-based visualization pipeline. By avoiding bus-transfers between CPU and GPU memory, all stages of this pipeline can be configured in real-time and by user interaction.

and detect untypical and potentially dangerous movement behavior. Scheepens et al. introduced interactive density maps by utilizing GPUs, encoding arbitrary attributes by modifying the kernel attributes and introduced an interactive approach for selecting subsets of trajectories [14].

Trajectory and Attribute Visualization: To visualize attributes of complex data, the concept of visual glyphs [15] can be used. Glyph-based visualizations have been applied to convey time-dependent data, such as time series, in the context of geographical maps [16], [17]. Tominski et al. have used a wall-like approach to visualize spatio-temporal data in a 2D geographical context [18], [19]. Nienhaus used dynamic glyphs to depict dynamics following visual art and graphic design principles [20]. In the context of movement trajectories, the visualization of trajectory attributes is an important functionality. To visualize attribute values together with the spatial and temporal information, Tominski et. al. used a stacking-based approach [21]. Andrienko et al. have applied a similar approach to the space-time cube to visualize attributes of trajectories following similar routes [22].

Air-Traffic Data: The visualization of movement trajectories is important in the context of air-traffic data. Hurter et al. have developed an interactive tool for the exploration of massive amounts of aircraft trajectories [23], using brushing and connected viewports. Also, image based approaches have been used to visualize and analyze aircraft trajectories: Hurter et al. have applied a temporal bundling approach to visualize trajectories and dynamic graphs [24], and extended the approach to develop visual-analytics methods for the analysis of large data sets of air-traffic trajectories [25], [26]. As a specific analysis task, they proposed a method to extract wind parameters from aircraft trajectories [27].

IV. VISUALIZATION PIPELINE ARCHITECTURE

To cope with massive spatio-temporal data, we propose a visualization pipeline, in which all stages (filtering, mapping, and rendering) are implemented on the GPU (Fig. 3). In particular, complex input data, consisting of attributed nodes and lines, is uploaded directly to the GPU, including all attributes available for interactive filtering and mapping. Geometry creation, attribute mapping, and rendering are then performed entirely on the GPU, resulting in an interactive visualization system, in which all options can be modified and applied in real-time. The concept can be implemented in terms of vertex, geometry, and fragment shaders:

1) The vertex shader is responsible for selecting and applying an appropriate mapping configuration for each vertex, i.e., a description of how attributes of a data item are mapped to geometric representations. Based on the input data of the vertex and additional information, such as the current selection and a level-of-detail parameter, the vertex shader selects a configuration, applies the mapping from data attributes to visual properties, and finally passes the mapping configuration on to the geometry shader. It also applies filtering by omitting the rasterization of primitives for items, which are not selected by the current filter options.

2) The geometry shader fetches a mapping configuration for each node or line segment and generates the appropriate geometry according to the selected mapping. This step is computationally expensive, yet it constitutes the flexibility of the approach. For example, it is possible to create completely different kinds of geometries, such as lines, tubes, or spheres, based on attributes of the input data, and this mapping can even be modified at runtime or by user interaction.

3) During rendering, the generated geometry is rasterized and an output image is generated. In addition, stylization and post-processing effects may be applied to improve the quality of the generated visualization artifact.

A. Interactive Filtering

Filtering of input data helps to focus a user's attention on the selected items, reduces visual clutter, and improves performance by reducing the amount of data that needs to be processed and visualized. For this, a user selects the subset of data, which is subsequently visualized, while the remaining items are either omitted or visualized less prominently. In our pipeline, three filtering concepts are supported: spatial filtering, temporal filtering, and attribute-based filtering. All three filtering options can be combined and controlled interactively by the user.

1) Spatial Filtering: Data items can be filtered by their geographic position, to restrict the analysis to a specific region-of-interest and thereby drastically reduce the amount of data that needs to be processed and visualized.

2) Temporal Filtering: This filtering method selects items by their temporal information, accepting items that lie within a certain time span or correspond to repeating time intervals, such as the current time of day. Such a time span is defined by user interaction (e.g., by displaying a virtual time line). Further, multiple time spans can be selected and used by the visualization methods, e.g., for a visual comparison of differences in density maps, or to highlight different temporal foci in a focus+context visualization.

3) Attribute-based Filtering: Items are filtered based on attribute values (e.g., using minimum and maximum values).

The resulting filtering options are passed on to the graphics card, where they are used to discard all items that do not apply to the selected filter rules. For a GPU-based implementation, this results in the geometry shader not emitting primitives for filtered data items, while the vertex for a data item must still be processed.

B. Hardware-accelerated Mapping

To visualize attributed trajectories, geometric representations of data items, and mapping of data attributes to their shapes and appearances are required. This can represent a challenging task in the case of complex data sets.

After filtering, geometric representations for all visualized data items have to be created, selecting the type of geometry, the parameterization of the geometry, and the configuration of how attribute values are mapped to geometric properties. This is called a *mapping configuration* in our system. For each individual data item, the mapping configuration is chosen based on the following criteria:

1) Input Data and Classification: Most importantly, the mapping configuration can depend on the data item itself, i.e., the values of its attributes. This can be used to classify data items and use different visualizations for each distinct class of items. For example, individual visualization configurations can be selected to visually differentiate between departing and arriving planes, or different types of aircrafts. Classifications can also be used to emphasize on specific sample points within a single trajectory, e.g., to identify curves or phases of acceleration.

2) Level-of-Detail: After a mapping configuration has been selected based on the data item itself, the mapping system automatically applies level-of-detail by selecting different configurations based on the current distance of the virtual camera to the visualized item. This provides the option to modify or even exchange the mapping configuration for each level-of-detail, e.g., to reduce visual details or decrease the tesselation factor for objects that have a high distance to the virtual camera.

3) Filtering: The results of the filtering step can also be used to influence mapping configurations. For example, for focus+context visualizations, different configurations can be used for selected (*focus*) and unselected items (*context*), or to distinguish between multiple defined foci.

4) Selection: Finally, the mapping configuration can also be influenced by user interaction, for example, by modifying or exchange mapping configurations to visually highlight those items, that are currently selected by the user.

C. Image Synthesis

This section briefly covers rendering techniques and concepts incorporated in our visualization pipeline to serve real-time rendering constraints, visual quality and provisioning of visualization artifacts.

1) Trajectory and Glyph Rendering: The central functionality of our visualization approach consists of the efficient rendering of 3D movement trajectories and their attributes. For this, several geometric primitives (lines, ribbons, and tubes) can be used. Additionally, individual sample points of trajectories can also be depicted by spheres, e.g., to emphasize on specific attribute values. To communicate the values of attributes, they can be mapped to visual properties, such as color, line width, or the diameter of tubes or spheres. This is supplemented by texture mapping and animation, which can for example be applied to convey the direction and speed of air planes using animated arrows.

Geometric primitives are rendered mainly with billboard techniques, using shading to create the impression of the specific shapes, such as tubes or spheres, while reducing the amount of geometry and tessellation that needs to be processed on the graphics board.

2) Generation of Density Maps: To support aggregated representations, density maps can be generated in real-time, using the filtering, mapping, and rendering pipeline as described before. The synthesis of density maps is performed using off-screen rendering in combination with alpha-blending. The resulting density maps can either be visualized directly, or be used as an input to additional analysis tools, e.g., for outlier detection or real-time edge bundling. Fig. 5 depicts an exemplary density map created by our visualization pipeline.

3) Post-processing: After rendering of the actual geometry, a number of post-processing steps are executed to apply additional effects on visualization artifacts and enhance the visual quality of the rendered images. In this step, image filters such as SSAO [28] and unsharp masking [29] are applied to enhance depth perception in massive 3D trajectory visualization and to ease differentiation of single trajectories. Also, image-based anti-aliasing [30] is applied to alleviate strong aliasing effects on the rendered edges. Another post-processing effect is responsible for applying highlighting effects on items that are currently selected by the user.

4) Server-based Rendering: To support the visualization, exploration, and analysis of massive trajectory data also on mobile devices and embedded web applications, we prototypically implemented server-side rendering [31]. Using concepts of web-perspective view services (WPVS) [32], the functionality of the visualization pipeline is exposed on a web-based programming interface. This can be used by client applications to create visualization artifacts and transfer the resulting images (e.g., color, depth, or ID buffers) over the network to the client application, where they are processed and displayed. This enables the implementation of interactive visual analysis applications also on thin clients.

V. APPLICATION EXAMPLES

Examples of visualizing massive air-traffic data in real-time are shown in Fig. 4. The following visualization and interaction techniques are supported:

A. Object Highlighting

Single or multiple object *highlighting* of specific scene objects (Fig. 4a) is a basic feedback functionality in interactive system, e.g., to emphasize objects that are selected by a user, or to communicate the results of a computation, such as outlier detection or general data base queries. As a result of configuration or interaction, the identifiers of all trajectories to be highlighted are stored in a highlighting buffer, and a specific mapping configuration is chosen during mapping (Sec. IV-B), when a highlighted trajectory is going to be rendered. Users can configure highlighting, ranging from differentiating objects visually, e.g., by assigning different colors, to rendering highlighted objects in greater detail or even displaying object details only on highlighted objects.

B. Temporal Focus+Context Visualization

Similar to [3], the mapping approach can also be used to create a temporal focus+context visualization. For that, all trajectories within a user-defined time interval are considered to be within the temporal focus, while the remaining trajectories represent the context. One or multiple temporal foci can be defined by manipulating a time interval in a dedicated user interface and selecting a distinct mapping configuration for the objects within that time interval. Another mapping configuration is used for the context (all unselected objects), and an additional configuration can be defined for objects in overlapping focus regions (i.e., objects that belong two more than one focus interval).

Temporal focus+context visualization enables users to visually differentiate and compare the trajectories in different time intervals and the respective context, e.g., to estimate the variation in numbers of trajectories. In Fig. 4e, two such focus regions are defined at disjunct time intervals. Trajectories within the focus areas are depicted in red and blue respectively, while trajectories in the temporal context are depicted as yellow dots. Fig. 4f shows a similar setting, but additionally visualizes trajectories within the overlapping time-interval using a third color (violet).

C. Density Maps

Density maps represent movement density, e.g., the traffic volume, for each spatial position. They can either be precomputed using a static configuration, or created in real-time, to reflect the current filter options and camera perspective selected by the user. Attribute mapping can be used to vary the influence of certain trajectories to the overall density, e.g., based on the speed or the weight class of an aircraft.

As a basic use case, density maps can be applied to assess the amount of traffic in a selected area and time interval (Fig. 5a). Further, they can be used to easily compare traffic volumes in different situations (Fig. 5b), such as specific time intervals or different planning scenarios, with each other. Another application can be the detection of outliers, which can be achieved by comparing individual trajectories to the regions of high density in the resulting density map.

D. Detail+Overview Visualization

Density maps can further be used for implementing detail+overview visualizations. Here, a density map constitutes the overview, and by selecting an area in the overview map, the contributing trajectories are highlighted in the detail view, which contains individual trajectory renderings.

E. Space-Time Cube

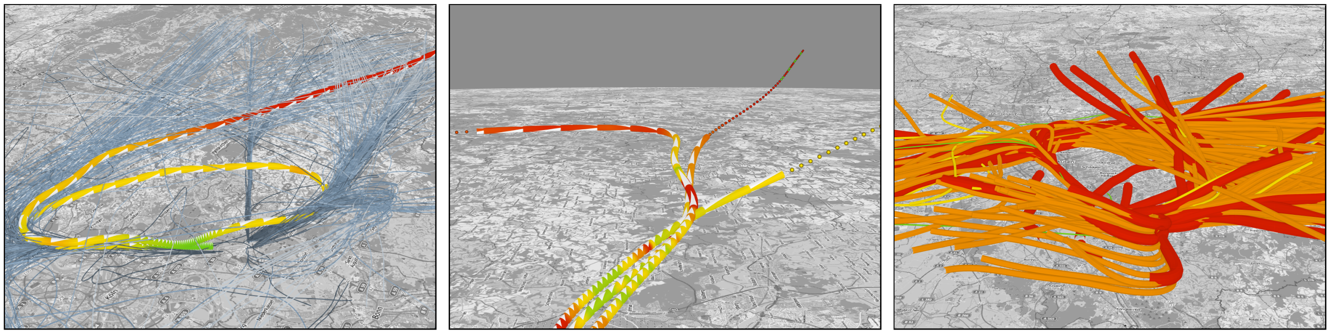
To show spatial and temporal relationships between trajectories at higher detail, e.g., to visually compare a number of trajectories, space-time cube visualization can be used. This can be achieved by applying an appropriate mapping configuration, which maps time of the trajectory to height of the geometry, as shown in Fig. 6. To avoid the problem of visual clutter, this should be combined with focus+context and filtering methods as described before. Apart from that, space-time cube visualizations can be freely combined with all mapping and visualization options shown before.

VI. CONCLUSIONS

This paper has presented a real-time animated visualization technique for massive 3D movement trajectories (e.g., air-traffic data). Its efficiency is achieved by a GPU-based implementation of all stages of the visualization pipeline (filtering, mapping, and rendering). The techniques can generally be applied as a key component for visual analytics tools that aim at interactive investigation, exploration, and analysis of large spatio-temporal data sets by expert users.

We have demonstrated its applications by several visualization and interaction techniques, such as 3D trajectory visualization, temporal focus+context, real-time computation of density maps, and space-time cube visualization. Using a data set of large, attributed air-traffic trajectories, we have explored the application of the described methods for developing visual-analytics tools in the context of ATM.

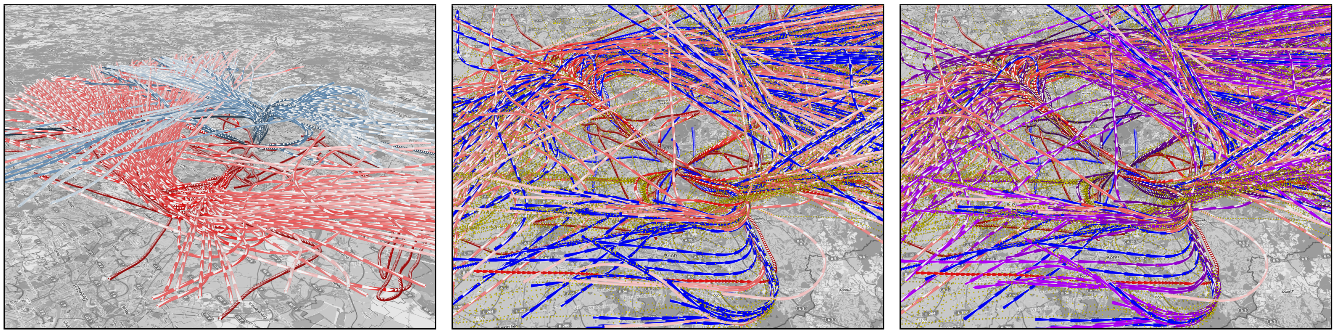
To increase the supported size of data sets, out-of-core and streaming algorithms should be applied. To still render such data sets at interactive frame rates, filtering and mapping also have to be extended, e.g., by applying automatic techniques that simplify the geometry of 3D trajectories. An application of stereoscopic 3D rendering, e.g., for 3D displays or immersive environments like CAVEs or projection domes, could be investigated to enhance the perception of 3D trajectories. Automatic analysis algorithms, e.g., to identify missed approaches or continous descent approach, should be



(a) Highlighting of a trajectory representing a missed-approach on an airport. The selected trajectory is rendered as a detailed tube, mapping speed to color. Unselected trajectories are displayed as thin lines, using a different color map.

(b) Dynamic mapping of per-sample-point attributes to visual properties. The trajectory is rendered as a tube in the focus area of the picture, in the rear, sample points are depicted by small spheres. Acceleration is mapped to color.

(c) Mapping of per-trajectory attributes to visual properties to visualize different aircraft types. The weight class of an aircraft is depicted by both diameter and color (from red for large aircrafts to green for light aircrafts.)



(d) Visualization of approaches (red) and departures (blue). Texture mapping and animation are used to communicate direction (arrow texture) and speed (texture stretching and animation speed).

(e) Temporal focus+context visualization to compare the trajectories of two different days. The two distinct temporal foci are depicted by red and blue tubes, the context is visualized by yellow dots.

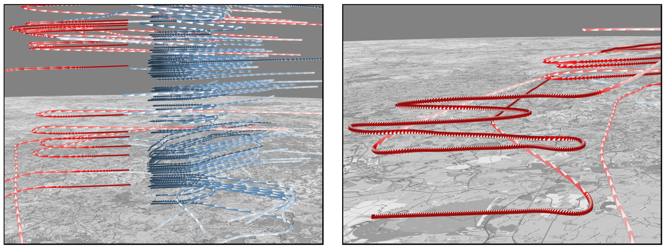
(f) Temporal focus+context visualization displaying two overlapping temporal foci (using red and blue for the two foci, violet for the overlapping area). The context is visualized by yellow dots.

Figure 4: Example applications using real-time animated 3D trajectory visualization.

(a) Aggregated view on the movements of air planes over a single day.

(b) Comparison of the distribution of movements between two days.

Figure 5: Visualization of 2D density maps.



(a) Temporal order of departing (blue) and arriving (red) air planes.

(b) Temporal relationship between a few numbers of trajectories.

Figure 6: Exemplary space-time cube visualization.

implemented into the GPU-based pipeline and exposed to users for real-time application. Finally, concepts such as the time wave [33] can be applied, to better reflect the linear and cyclic characteristics of temporal data and to support temporal exploration at finer grades.

ACKNOWLEDGMENTS

This work was funded by the German Federal Ministry of Education and Research (BMBF) in the InnoProfile Transfer research group ”4DnDVis”. We also wish to thank Deutsche Flugsicherung GmbH for providing the used data set.

REFERENCES

[1] N. Andrienko and G. Andrienko, *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, 2005.

[2] G. Andrienko, N. Andrienko *et al.*, "Space, time and visual analytics," *International Journal of Geographical Information Science*, vol. 24, no. 10, pp. 1577–1600, 2010.

[3] A. Carvalho, A. A. de Sousa, C. Ribeiro, and E. Costa, "A temporal focus + context visualization model for handling valid-time spatial information," *Information Visualization*, vol. 7, no. 3, pp. 265–274, 2008.

[4] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti, "Interactive visual clustering of large collections of trajectories," *IEEE Symposium on Visual Analytics Science and Technology*, pp. 3–10, 2009.

[5] G. Andrienko and N. Andrienko, "Interactive cluster analysis of diverse types of spatiotemporal data," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 2, pp. 19–28, 2010.

[6] N. Elmqvist and P. Tsigas, "A Taxonomy of 3D Occlusion Management for Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, pp. 1095–1109, 2008.

[7] T. Hägerstrand, "What about people in regional science?" *Papers of the Regional Science Association*, pp. 7–21, 1970.

[8] M.-J. Kraak, "The space-time cube revisited from a geovisualization perspective," *Proc. 21st International Cartographic Conference*, pp. 10–16, 2003.

[9] N. Andrienko, G. Andrienko, and P. Gatalsky, "Visual data exploration using space-time cube," *21st International Cartographic Conference*, pp. 1981–1983, 2003.

[10] M.-J. Kraak and A. Koussoulakou, "A visualization environment for the space-time-cube," in *Developments in Spatial Data Handling: 11th International Symposium on Spatial Data Handling*. Springer, 2005, pp. 189–200.

[11] P. O. Kristensson, N. Dahlbäck, D. Anundi, M. Björnstad, H. Gillberg, J. Haraldsson, I. Mårtensson, M. Nordvall, and J. Ståhl, "An evaluation of space time cube representation of spatiotemporal patterns," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 4, pp. 696–702, 2009.

[12] I. Kveladze, M.-J. Kraak, and C. P. van Elzakker, "A methodological framework for researching the usability of the space-time cube," *The Cartographic Journal*, vol. 50, no. 3, pp. 201–210, 2013.

[13] N. Willems, H. van de Wetering, and J. van Wijk, "Visualization of vessel movements," *Computer Graphics Forum*, vol. 28, no. 3, pp. 959–966, 2009.

[14] R. Scheepens, N. Willems, H. van de Wetering, and J. J. van Wijk, "Interactive Density Maps for Moving Objects," *IEEE Computer Graphics and Applications*, vol. 32, no. 1, pp. 56–66, 2012.

[15] A. M. MacEachren, *How Maps Work: Representation, Visualization and Design*. Guilford Press, 1995.

[16] C. Tominski, P. Schulze-Wollgast, and H. Schumann, "3D Information Visualization for Time Dependent Data on Maps," *Ninth International Conference on Information Visualisation (IV'05)*, pp. 175–181.

[17] T. Sidharth and A. Hanson, "A 3D Visualization of Multiple Time Series on Maps," *14th International Conference Information Visualisation*, pp. 336–343, 2010.

[18] C. Tominski and H.-J. Schulz, "The Great Wall of Space-Time," in *Workshop on Vision, Modeling & Visualization (VMV)*. Eurographics Association, 2012, pp. 199–206.

[19] ——, "A Wall-Like Visualization for Spatio-Temporal Data," Workshop GeoVisual Analytics, Time to Focus on Time, at GIScience, 2012.

[20] M. Nienhaus and J. Döllner, "Depicting dynamics using principles of visual art and narrations," *IEEE Computer Graphics and Applications*, vol. 25, no. 3, pp. 40–51, 2005.

[21] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko, "Stacking-Based Visualization of Trajectory Attribute Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2565–2574, 2012.

[22] G. Andrienko, N. Andrienko, H. Schumann, and C. Tominski, "Visualization of Trajectory Attributes in Space–Time Cube and Trajectory Wall," in *Cartography from Pole to Pole*. Springer, 2014, pp. 157–163.

[23] 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.

[24] C. Hurter, O. Ersoy, S. Fabrikant, T. Klein, and A. Telea, "Bundled visualization of dynamic graph and trail data," *IEEE Transactions on Visualization and Computer Graphics*, 2013.

[25] T. Klein, M. van der Zwan, and A. Telea, "Dynamic multiscale visualization of flight data," in *VISAPP 2014*, 2014.

[26] C. Hurter, S. Conversy, D. Gianazza, and A. Telea, "Interactive image-based information visualization for aircraft trajectory analysis," *Transportation Research Part C: Emerging Technologies*, 2014.

[27] C. Hurter, R. Alligier, D. Gianazza, S. Puechmorel, G. Andrienko, and N. Andrienko, "Wind parameters extraction from aircraft trajectories," *Computers, Environment and Urban Systems*, 2014.

[28] L. Bavoil and M. Sainz, "Screen space ambient occlusion," *NVIDIA*, vol. 6, 2008.

[29] T. Luft, C. Colditz, and O. Deussen, "Image enhancement by unsharp masking the depth buffer," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1206–1213, Jul. 2006.

[30] T. Lottes, "Fxaa," Available from http://developer.download.nvidia.com/assets/gamedev/files/sdk/11/FXAA_WhitePaper.pdf, 2009, [Accessed 8 June 2014].

[31] J. Klimke, B. Hagedorn, and J. Döllner, “Scalable multi-platform distribution of spatial 3d contents,” in *ISPRS 8th 3D GeoInfo Conference & WG II/2 Workshop*, ser. ISPRS Annals, U. Isikdag, Ed., vol. II-2/W1, 2013, pp. 193–200.

[32] “Web View Service Discussion Paper, Version 0.3.0,” OGC Discussion Paper, OGC 09-166r2, 2010.

[33] X. Li and M.-J. Kraak, “The time wave. a new method of visual exploration of geo-data in time–space,” *The Cartographic Journal*, vol. 45, no. 3, pp. 193–200, 2008.