HAL
open science

Visualization, Selection, and Analysis of Traffic Flows

Roeland Scheepens,, Christophe Hurter, Huub van de Wetering, Jarke J. van Wijk

To cite this version:

Roeland Scheepens,, Christophe Hurter, Huub van de Wetering, Jarke J. van Wijk. Visualization, Selection, and Analysis of Traffic Flows. IEEE Transactions on Visualization and Computer Graphics, 2015, PP (99), 10.1109/TVCG.2015.2467112 . hal-01206660

HAL Id: hal-01206660

https://enac.hal.science/hal-01206660

Submitted on 6 Oct 2015

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.

ResearchGate

See discussions, stats, and author profiles for this publication at: http://www.researchgate.net/publication/282047238

Visualization, Selection, and Analysis of Traffic Flows

ARTICLE · SEPTEMBER 2015

DOI: 10.1109/TVCG.2015.2467112

READS

12

4 AUTHORS, INCLUDING:

Roeland Scheepens

Technische Universiteit Eindhoven

9 PUBLICATIONS **56** CITATIONS

SEE PROFILE

Christophe Hurter

Ecole Nationale de l'Aviation Civile

79 PUBLICATIONS **231** CITATIONS

SEE PROFILE

Available from: Roeland Scheepens
Retrieved on: 02 October 2015

Visualization, Selection, and Analysis of Traffic Flows

Roeland Scheepens, Christophe Hurter, Huub van de Wetering, and Jarke J. van Wijk

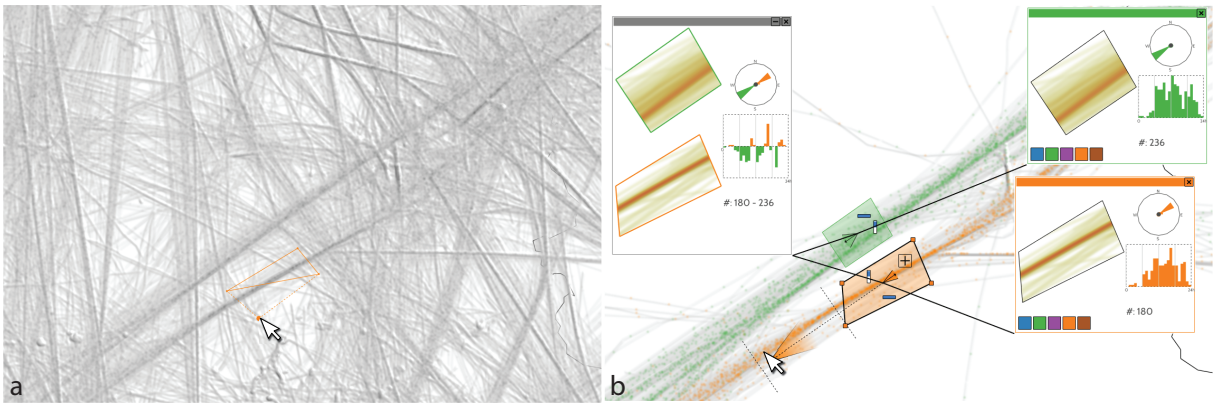

Fig. 1. (a) Traffic flows are visualized using a density map and a particle system. They can be selected using our novel selection widget, and are visible through colored particles. (b) Two traffic flows of aircraft moving in opposite directions have been selected. The green traffic flow is moving south-west, and the orange traffic flow is heading north-east. Their dynamics can be explored and compared through movable windows, which show distributions of direction, and density over time. The gray window shows the difference between the two traffic flows.

Abstract—Visualization of the trajectories of moving objects leads to dense and cluttered images, which hinders exploration and understanding. It also hinders adding additional visual information, such as direction, and makes it difficult to interactively extract traffic flows, *i.e.*, subsets of trajectories. In this paper we present our approach to visualize traffic flows and provide interaction tools to support their exploration. We show an overview of the traffic using a density map. The directions of traffic flows are visualized using a particle system on top of the density map. The user can extract traffic flows using a novel selection widget that allows for the intuitive selection of an area, and filtering on a range of directions and any additional attributes. Using simple, visual set expressions, the user can construct more complicated selections. The dynamic behaviors of selected flows may then be shown in annotation windows in which they can be interactively explored and compared. We validate our approach through use cases where we explore and analyze the temporal behavior of aircraft and vessel trajectories, *e.g.*, landing and takeoff sequences, or the evolution of flight route density. The aircraft use cases have been developed and validated in collaboration with domain experts.

Index Terms—Moving Object Visualization, traffic flows, interaction

1 Introduction

Moving objects such as cars, vessels, aircraft, or pedestrians do not move at random, but collectively form patterns. These collective patterns, or dynamic collective behavior [4], may be formed in two ways [12]: By groups that share some functional relationship, such as groups of animals travelling together, or by cohorts, which have some other factor in common, such as aircraft that have the same destination, or vessels that use the same shipping lane. In this paper, we are interested in analyzing the latter patterns, which we call *traffic flows*. A traffic flow is represented by a set of trajectories. This gives rise to a number of challenges: How to visualize an overview of all trajectories such that a user can easily find traffic flows of interest; how to select these traffic flows; how to analyze their dynamics, *i.e.*, their behaviour over time; and how to compare multiple flows.

Analyzing trajectories of moving objects is of great interest to extract temporal patterns and to understand past events. Air traffic management deals with moving objects, namely aircraft, which follow flight routes through sectors (volumes) that subdivide the airspace under its control. For improved flight safety and flight route optimization, flight routes and sectors may evolve over time. Defining these optimizations properly is complex, and relies on a deep understanding of the dynamics of the air traffic. To facilitate this, we investigate how to increase the understanding and aid the analysis of flow dynamics using an interactive visual approach.

Recorded data may have poor semantics when only the location of the moving objects is available at a given time, but additional information can be derived with simple algorithms, for example: speed, acceleration, direction. After this additional information processing, trajectories can be analyzed to extract relevant insights. Interactive visualization systems can leverage human visual analytical skills to get these insights. Since trajectories of moving objects are directed, the direction information is highly relevant and important to display. Most existing systems use arrows to show direction [5], leading to cluttered views that hinder data exploration. Other solutions investigated the use of color gradients [19]; this requires specific visual mappings, and subsequent color blending gives issues when gradients overlap. Some systems employ animated textures to show direction [7], requiring a minimum trajectory width, which is not suitable for large data sets

- *Roeland Scheepens, Huub van de Wetering, and Jarke J. van Wijk are with the Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands, E-mail: R.J.Scheepens, H.v.d.Wetering, J.J.v.Wijk@tue.nl.*
- *Christophe Hurter is with the Interactive Computing Laboratory (LII) of the French Civil Aviation University (ENAC) in Toulouse, France, E-mail : christophe.hurter@enac.fr*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

with many entangled traffic flows. In this paper, we propose the use of animated particles. Particle systems are an under exploited visualization technique that show the spatial extent and direction of a trajectory through moving particles and can provide strong cues with little clutter. Even if trajectories overlap, particle movements remain visible and indicate trajectory directions. Moreover, combined with particle density indicating trajectory densities, traffic flows become visible. As a drawback, a particle system requires animation with a high frame rate and interactive response time. The latter requirements are especially challenging for large moving object data sets.

Our overall contribution is to provide an integrated set of visualization and interaction techniques for investigating the dynamics of traffic flows. In our final design, we use a density map of the trajectories combined with moving particles. We developed a new multi-dimensional selection widget for traffic flows. This interaction technique allows geographical selection of traffic flows, possibly filtered on direction and additional dimensions such as altitude or velocity. Such interaction helps to better highlight subsets of the traffic flows and to perform qualitative visual comparison in terms of direction and density. Furthermore, we provide additional visualizations to show and compare selections and their dynamics. The presented work illustrates how a particle system can be a great asset to explore a multi-dimensional data set of moving objects and how we developed visualization and interaction techniques to take full advantage of them. To support our claim, we present a number of case studies where our approach helps to retrieve insight from data sets of different kinds of moving objects: vessels and aircraft.

In Section 2 we discuss related approaches. In Section 3 we explore our problem description by discussing input data, the target users and their requirements, and in Section 4, we discuss a set of typical tasks. Following this, we discuss our approach in Sections 5 (Visualization), 6 (Selection), and 7 (Analysis). We demonstrate and validate our approach according to a number of use cases using aircraft and vessel data in Section 8. And finally, we discuss our work, draw conclusions and discuss future work in the final section.

2 Related Work

Much prior work already exists regarding spatio-temporal data exploration [6]. In the following, we focus on work related to moving object visualization and exploration and outline our improvements.

Andrienko and Andrienko [1] investigate visualizing and comparing the variation in spatially distributed time-series data, which they call behavior. The distribution of these behaviors is displayed on a map using glyphs that show the time graph of the time-varying attribute for each area. This technique is, however, based on time-varying data that is tied to a predefined area, whereas we allow the user to dynamically select traffic flows.

FromDaDy [22] is a tool that allows the exploration of trajectory data through brushing, and picking and dropping the selections into juxtaposed views. We improve on this by introducing a novel selection widget that allows the user to intuitively select traffic flows more precisely based on direction, and potentially other attributes. Tominski *et al.* [33] visualize time varying attributes of trajectories by stacking trajectory bands that encode attribute values by color in 3D space.

Andrienko *et al.* [3] present a visual analytics approach to find places of interest in movement data by clustering related movement events, such as low speed events to find locations with traffic jams. The temporal patterns of the movement data in these places of interest can then be studied using spatio-temporal aggregation. In our case we do not use automated methods to find places of interest as our users already know where to look and instead offer a visual method to select traffic flows of interest.

The exploration and visualization of Origin-Destination (OD) data is a growing research topic [36]. OD data is movement data for which only the origin and destination points are stored with additional attributes of the trips. Ferreira *et al.* [14] present a tool in which queries on OD data can be defined visually using graphical widgets. The query results can be visually explored and further refined. Guo *et al.* [15] automatically select and extract flow patterns from OD data, and visualize these patterns using arrow glyphs on a map. Though improvements have been proposed [9], resolving clutter in flow mapping remains a challenging problem. In OD exploration and visualization, however, the actual route and directions of the trajectories are not important, whereas in our approach we look at traffic flows where the origins and destinations of the individual trajectories are of less importance.

Krüger *et al.* [27] present *TrajectoryLenses*, a system that uses lenses to support visual, set-based filter expressions to select trajectories. It supports three types of lenses: origin, destination, and waypoint. These lenses can be grouped with set operations to create more complex queries. We follow a similar Focus+Context [11] technique where we visualize our selections and their dynamics within the context of all trajectories. Our selections can also be combined using set operations. Our selection widget, however, is more flexible. The selection area is user-defined and the selection can be filtered on multiple attributes.

Selection of lines by their direction or angle is also relevant in other domains. For example, Hauser *et al.* [17] introduce *angular brushing* in parallel coordinate plots to select records by their slope, or the relation between axes.

Van den Elzen and van Wijk [34] present a method to explore both the structure and the attributes of multivariate networks by creating selections of interest and combining a visualization of the selections in the network with a high level, infographic-style overview, showing the structure and multivariate attributes of the selections. We follow a similar approach, but we select traffic flows rather than locations. In our method, we can also generate high level, infographic-style visualizations based on user-made selections.

Muigg *et al.* [29] present a method to address clutter and overdraw in dense line plots. First a tensor field is generated with a distribution of line orientations for each pixel. Line orientations are then visualized by applying anisotropic diffusion to a noise texture. We get a similar effect using our method, however, animated particles perform better in areas where multiple directions overlap.

Willems *et al.* [35] visualize large numbers of moving object trajectories using density maps. Scheepens *et al.* [31] extend this by using multivariate filters. Density maps of subsets of the data can then be interactively generated. These density maps can be aggregated or visually composed using a variety of operators. An overview can be given of how traffic evolves over time, however, it requires much interaction, and traffic flows cannot be selected or investigated seperately. Neither can users see the direction of the traffic flows. In a further extension to this work, Scheepens *et al.* [30] use a scripting language to define more complex, composite density maps. This method can be used to extract and visualize dominant traffic flows in the data, but still cannot show direction. We also use density maps, but only to show an overview of the spatial distribution of the traffic flows.

Blaas *et al.* [7] use animated textures to display directionality in connected graphs to explore time series of an extended state-graph. Our work differs in many ways, but uses a similar visual technique to display flow direction.

Flow visualization is a large field of research in which many methods have been developed to visualize properties of flow fields, such as streamlines, glyphs, and textures [28]. However, such flows differ fundamentally from our data in that they are fields with (time-varying) properties tied to location, whereas our time-varying properties are tied to moving objects, whose trajectories may overlap.

3 Problem Description

In this section we describe our problem area in more detail. We start by describing our input data and the type of movement data we focus on. Following this, we discuss our target users and their requirements.

3.1 Data

Data that can be explored using our approach consists of a large set of trajectories of moving objects. We describe a trajectory of a moving object o by a state $\boldsymbol{\alpha}^{o}(t)$. A state is a tuple containing at least attributes such as position $\mathbf{p}_o(t)$, velocity $v_o(t)$, and heading $h_o(t)$, but may also

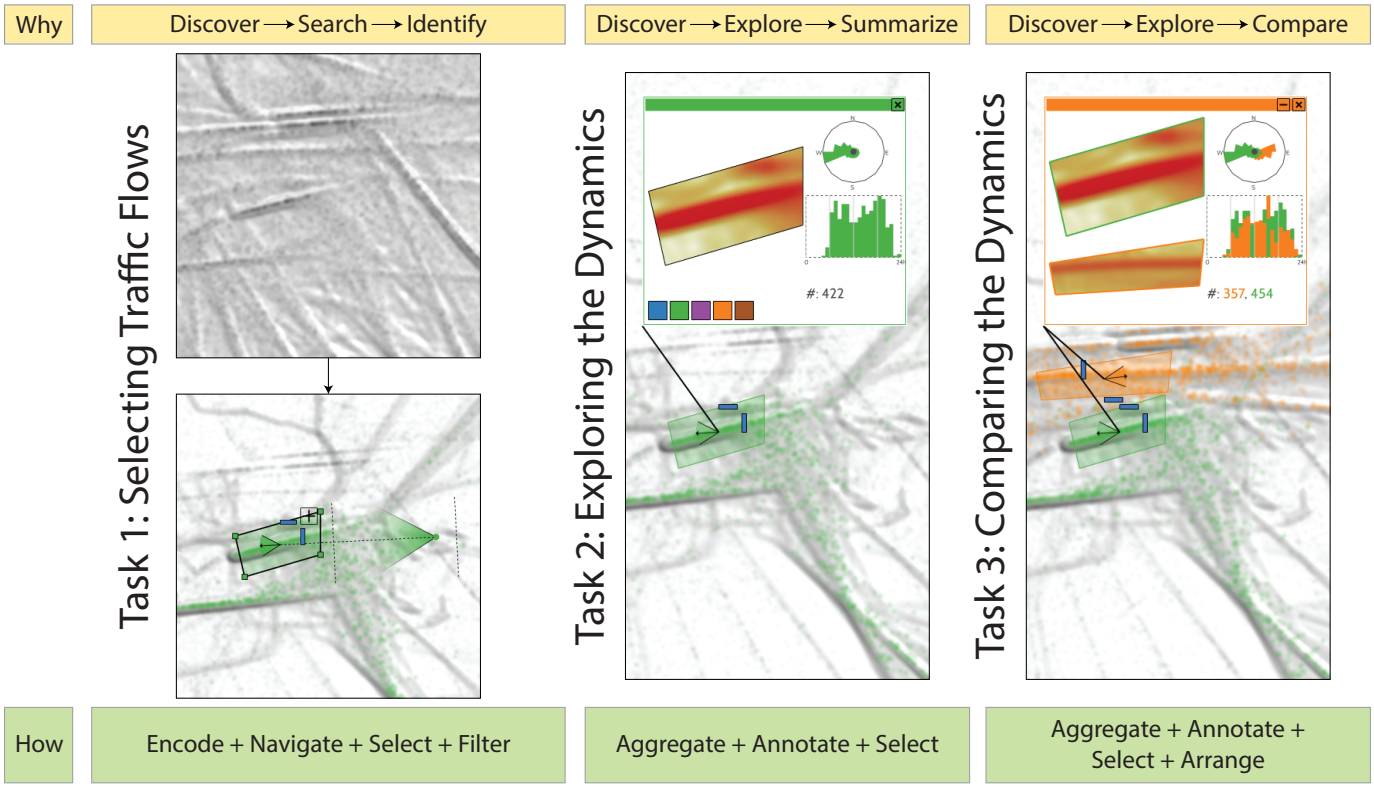

Fig. 2. An overview of our tasks, based on the typology of visualization tasks of Brehmer and Munzner [8]. In Task 1 the user can discover, search for, and identify traffic flows when navigating through a visualization of traffic flows encoded using a density map and a particle system. Traffic flows can be selected, and the selections can be filtered. In Task 2 the user can discover, explore, and summarize the dynamics of a traffic flow selected by Task 1. In Task 3 the user can compare multiple traffic flows by arranging multiple windows produced by Task 2 on top of each other.

contain additional attributes such as the altitude of an aircraft $a_o(t)$, or the type of a vessel τ_o.

We focus on movement data of objects that are free to move around in space, but are constrained by some rules that encourage dynamic collective behaviour, such as aircraft that follow flight routes, or vessels that follow shipping lanes. We do not focus on moving objects that move around freely without such rules, such as animal movements, or players in a football match. Our approach may also work for objects that are not free to move around in space, but follow a predefined track, such as trains on a railway track, but the highly constrained nature of these movements allow for simpler selection methods, such as brushing.

3.2 Requirements

Our approach is intended for analysts that want to investigate traffic flows. In our design we are mainly inspired by air traffic analysts. These analysts perform traffic flow analysis and are mostly air traffic controllers with extensive knowledge of existing flight rules and the structure of the airspace. Traffic flow analysis is required to optimize flight routes and sectors in the airspace structure, but can also provide important input for efficiently assigning the required number of air traffic controllers during a day. Gaining deep understanding of flow locations and their dynamics enables the extraction of even more information. To facilitate this, the user needs to see an overview of the flows and their departure/arrival locations; in addition, multidimensional filtering to focus on a specific flow in space and time, information on the number of aircraft per hour in an area, and traffic flow comparison should all be available.

Based on discussions with air traffic analysts, we have formulated the following requirements for our approach:

R1 **Visualization**. The user should be enabled to see the direction of traffic flows while looking at the overview of all trajectories.

R2 **Selection**. The user should be enabled to dynamically, and visually select any, and multiple, traffic flows of interest using simple interaction techniques.

R3 **Analysis**. The user should be enabled to investigate and compare the dynamics of the selected traffic flows, *i.e.*, how the flows evolve over time.

4 Task Analysis

Initially, we tried to find and visualize potentially interesting areas using automated methods inspired by [3]. We automatically found areas with a high dispersion of trajectory directions by computing the entropy [18] in a fine grid of cells. All points with a sufficiently high entropy are considered interesting and are clustered. The resulting clusters are used to define a Voronoi space division and for each Voronoi cell, relevant information is visualized, such as the distribution of directions, and the traffic density over time. Our users, however, generally know which areas they want to investigate, and preferred to define the areas of interest themselves.

Using the requirements stated in Section 3.2, we now identify several user tasks and describe these tasks according to the typology of visualization tasks by Brehmer and Munzner [8]. To investigate the usage of space, such as flight routes for aircraft, or shipping lanes for vessels, the user needs to be enabled to investigate the dynamics of traffic flows, *i.e.*, how the traffic flows behave over time. We have identified the following tasks–see Figure 2.

Task 1: Selecting traffic flows.
A user must be enabled to visually *discover* traffic flows. The user does this by *searching* for these traffic flows and *identifying* them. Traffic flows and their locations may or may not be known beforehand. To support the discovery process, we *encode* the traffic flows and their flow directions using a density map and a particle system, and allow

the user to *navigate* the map. The user may then *select* the traffic flow and *filter* the selection to *identify* the desired traffic flow.

Task 2: Exploring the dynamics.
In this *discovery process*, the user must be enabled to *explore* the dynamics of a traffic flow selected by Task 1, and *summarize* the dynamic attributes of all trajectories in the traffic flow. We do this by *aggregation* and we *annotate* the dynamics of the selected traffic flow in a window that the user can arrange within the visualization space. Furthermore, the user can browse through the aggregated time bins in this window by *selecting* individual bins.

Task 3: Comparing the dynamics.
A user must also be enabled to *discover* the relationship between multiple traffic flows selected by multiple instances of Task 2. The user must *explore* and *compare* the dynamics of these traffic flows. Similarly to Task 2, we do this by *aggregating* and *annotating* the dynamics of the selected traffic flows in windows. To enable the *comparison* of selected traffic flows, the time bin selection in all windows is linked. Additionally, the user can *compare* multiple traffic flows by *arranging* their respective windows on top of each other, which automatically *aggregates* the visualization of both windows.

Task 4: Infographic-style visualizations.
In the final task, the user may want to create an infographic-style visualization, in which case a visualization is *produced* for a third party. For this task, any combination of the previous tasks may serve as input. While this task does not follow from the requirements, we find it interesting to explore, nonetheless.

Overview The user can visualize, select, and analyze traffic flows using our approach as follows: We show the user an overview of the traffic flows in a user-defined time window by combining a density map [31] with animated particles. The density map shows the spatial overview of the trajectory data, while the particles show the direction of the traffic flows. The user can then interactively select traffic flows using a selection widget. We use the shading of the density map, and the color channel is used to visualize trajectory selections in the animated particles. To analyze the dynamics of the selected traffic flows and compare the dynamics of multiple selected traffic flows, the user can pop up windows for gaining insight in the dynamics of the selected traffic flows.

5 Visualization: Particles

The user should be enabled to see the direction of traffic flows while looking at the overview of all trajectories (R1). We have chosen to show these traffic flows using an animated particle system. There are several different methods to visualize the direction in an aggregated visualization of a large number of trajectories, such as color maps [19] or glyphs, for example arrows [2], which may be animated [10]. By encoding direction using color maps, blending issues arise and it is more difficult to use color to encode other attributes. By encoding direction using glyphs, a large amount of clutter is introduced, making the visualization harder to read, even if large traffic flows were bundled [19]. Instead, we have opted for animation, using a particle system, as we have found this does not interfere with other visual parameters, while it allows the user to clearly see and separate dominant traffic flows in the data–see the supplemental video. We have found that this does require a smooth animation at a high frame rate. We initially feared having animated particles would distract the user, but we have found this is not the case.

Having chosen for a particle system, we now derive the following new requirement from requirement R1: The particles should be visualized and animated such that the user can observe and distinguish traffic flows, even in areas where traffic flows intersect. Our particle system is produced as follows. First, we generate, for each trajectory $\boldsymbol{\alpha}^{\boldsymbol{o}}(t)$, a set of particles geographically equally spaced along the trajectory. Their spacing needs to be such that the particle flows are dense enough to be able to observe traffic flows at all times, and sparse enough to be able to see particles moving along a traffic flow and to distinguish crossing traffic flows. The particles are then cycled over the trajectories with some speed defined in screen space. To avoid distracting regular patterns, some jitter in time and space may be required,

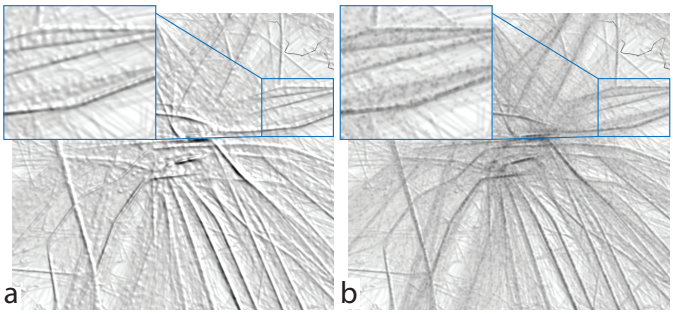

Fig. 3. (a) The particles rendered as part of the shading of the density map, and (b) the particles rendered as Gaussian bells.

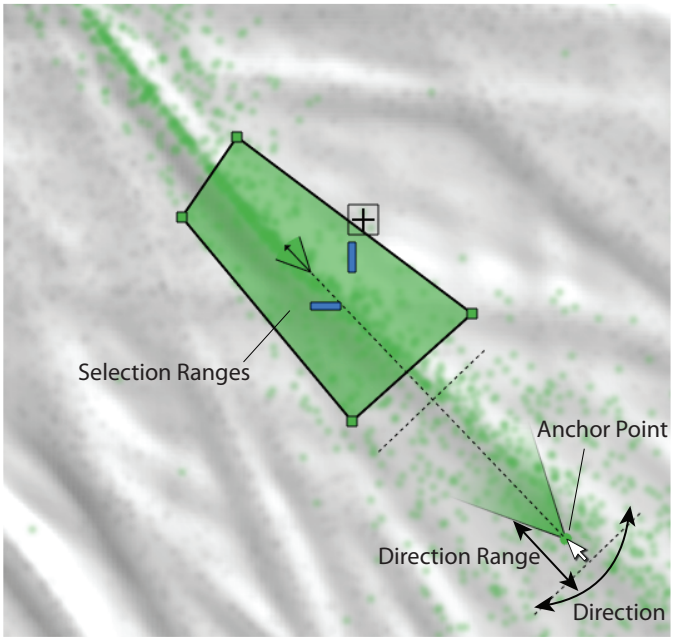

Fig. 4. The user can create a selection area by clicking a polygonal area on the map. An initial direction range selection is estimated, which can be changed by the user in an intuitive way, by dragging the anchor point. The glyph in the center of the selection area shows the current selection ranges.

however, we have found that in our data sets this was not needed. To render the particles themselves, we explored two possibilities–see Figure 3. First, we can render the particles as part of the density maps, *i.e.*, as part of the shading similar to [31]. Second, we can render the particles as blended Gaussian bells with finite support. Rendering the particles in the shading makes it harder to use color to visualize additional attributes. Moreover, it appeared harder to interpret than the blended Guassian bells–see the supplemental video. Therefore, the particles are rendered as black or colored Gaussian bells, which are alpha blended using some weight p_α. Both the particle spacing and the weight are automatically estimated based on the number of trajectories per km^2 in the data set such that the particles reveal clear movement patterns without dominating the view. For example, in the aircraft data set shown in Figures 3 and 4, a particle spacing of approximately 40 km is used. The particle spacing remains constant while zooming, however, the particle size increases.

Because particles are generated for each trajectory, the particle system also naturally shows traffic density. We explicitly asked all domain experts we consulted what they thought the particles represent. Even without any further explanation, none of the domain experts confused a particle as representing a single aircraft or vessel.

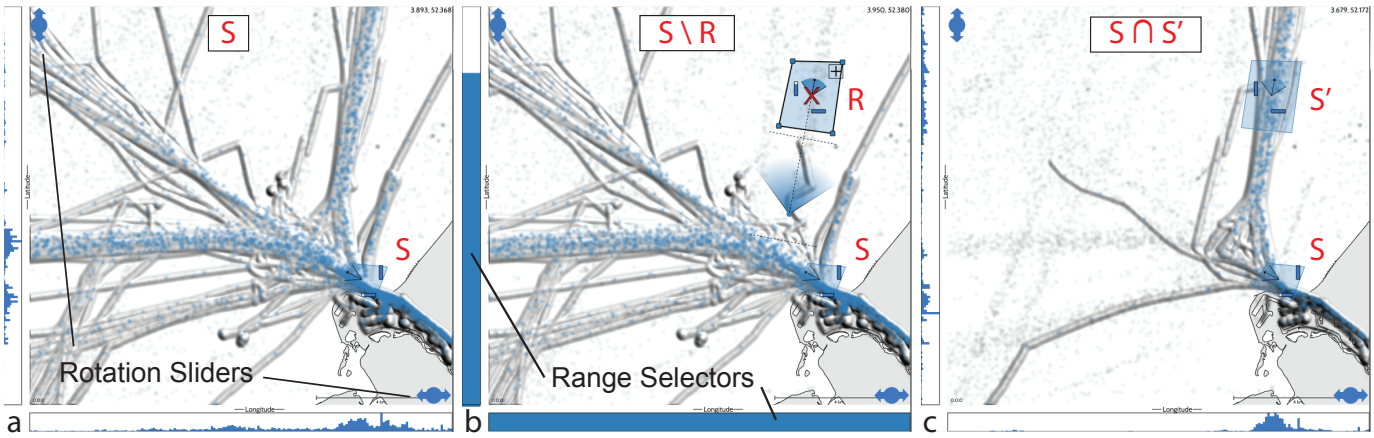

Fig. 5. Selections of traffic flows of vessels near the harbor mouth of the port of *Rotterdam*. (a) Vessels leaving the harbor have been selected (S), (b) vessels leaving the harbor (S) except vessels heading north (R) have been selected, and (c) vessels leaving the harbor (S) that head north (S') have been selected.

A slider can be used to change the particle weight p_α of non-selected particles, which also enables the user to hide non-selected particles by setting their weight to 0. To support zooming, the radius of the particles is defined in world space.

The velocity of the particles can be used to encode attributes of the moving objects they represent. We have experimented with attributes such as the actual velocity of the moving objects, and other attributes such as altitude. Encoding the actual velocity of the moving objects appears to distract the user and reduces the visual strength of traffic flows as too many particles move in different velocities in the same area. Mapping the altitude of aircraft to a limited number of discrete bins with particle velocities, using the principle that objects that are closer, *i.e.*, higher, to the camera, appear to move faster, seems to allow the user to separate traffic flows by altitude. The effect, however, is not pre-attentive and makes separating traffic flows in general slightly harder–see the supplemental video. We therefore use a constant particle speed, which can be changed by the user if needed. Additionally, we tried applying the semantic depth of field technique [26], by allowing the user to *focus* on a specific attribute value or range of values, such as a range of aircraft altitudes. The particles with attribute values within the focus range were rendered as normal, while particles with attribute values outside the focus range were blurred. The effect, however, was too subtle and does not allow the user to properly focus on the traffic flows in range, while maintaining an overview of the general traffic flow–see the supplemental video.

Since moving objects can have multiple dynamic or static attributes, we have chosen to map these attributes to additional dimensions, such that the user can change axes (by rotating the view [22]), to explore these attributes in a similar way as ScatterDice [13]. Another way to map these additional attributes would be by coloring particles, however, we already use color to mark selections of traffic flows. By dragging the sliders shown in Figure 5a, the user can control the rotation over the attributes. Lastly, while in the rotated view, the user can also rotate over the spatial z-axis to view the particles from different angles.

6 Selection

The user should be enabled to dynamically and visually select any, and multiple, traffic flows of interest using simple interaction techniques (R2). We enable the user to create a selection S, a set of trajectories representing traffic flows of interest, using novel selection widgets. The user can draw a polygonal selection area by clicking at least three points on the screen. All trajectories that visit this area are now selected. We have chosen a polygonal area as opposed to a circular area or brushing a line to give the user the flexibility to define their own areas of interest, which may be later used to analyze traffic flows as described in Section 7. The selected area can be modified interactively, by dragging one of the vertices of the polygon or by moving the polygon in its entirety. The selection S can be further refined by filtering a range of directions, such that only trajectories within the filtered direction range in the selection area are selected. In Figure 4 such a selection S is shown, with our selection widget that allows the user to intuitively filter a range of directions. The user can change the direction filter by moving the anchor point, rotating it around the area to rotate the direction filter and dragging the anchor point closer to or further from the selection area to, respectively, widen or narrow the range of the filter–see the supplemental video. The selection is updated immediately after the user releases the mouse button. To reduce the amount of user interaction required, we divide the trajectories in the selected area into segments of equal length. We then apply Principal Component Analysis (PCA) to the set of displacement vectors of these segments to find the principle direction of the trajectories. Finally, we align the widget direction range to this principal direction. This allows us to define a sensible direction range filter immediately after the selection area has been created. Ranges from additional attributes can be filtered using range selectors to the left or at the bottom of the screen–see Figure 5b, or by rotating the view and changing the selection area there. These selection ranges are also visualized using small, blue rectangles in the selection glyph–see Figure 5. In Figure 9b we see an example of aircraft traffic rotated into the altitude view. In this view, the user can interact with the selection areas to change their altitude filters–see supplemental video.

To enable the user to create more complicated selections, we support a compound selection S, *i.e.*, a selection created using multiple selections S_i–see Figure 4 and the supplemental video. Additionally, we support *inverse* selections R_i to remove trajectories from the compound selection S. A compound selection S is then defined by

$$S = \bigoplus S_i \setminus \bigcup R_i \tag{1}$$

where $\bigoplus$ can be configured to be $\bigcap$ or $\bigcup$. This allows the user for instance to define selections where all trajectories going through a selected area S are selected except those that go through another selected area R (Figure 5b), or where all trajectories that go both through selection area S and through selection area S' are selected (Figure 5c).

The user can also create multiple, independent compound selections. These selections are colored using a qualitative ColorBrewer color map [16]. Selected trajectories are visualized by coloring their particles using their associated selection color. To improve visibility, selected particles are rendered on top and with a higher particle weight p_α. The user can choose to only render the selected trajectories in the density map by selecting the "render only selected trajectories" option. Additionally, the user can choose to hide the selection widgets of all colors, or just of one color.

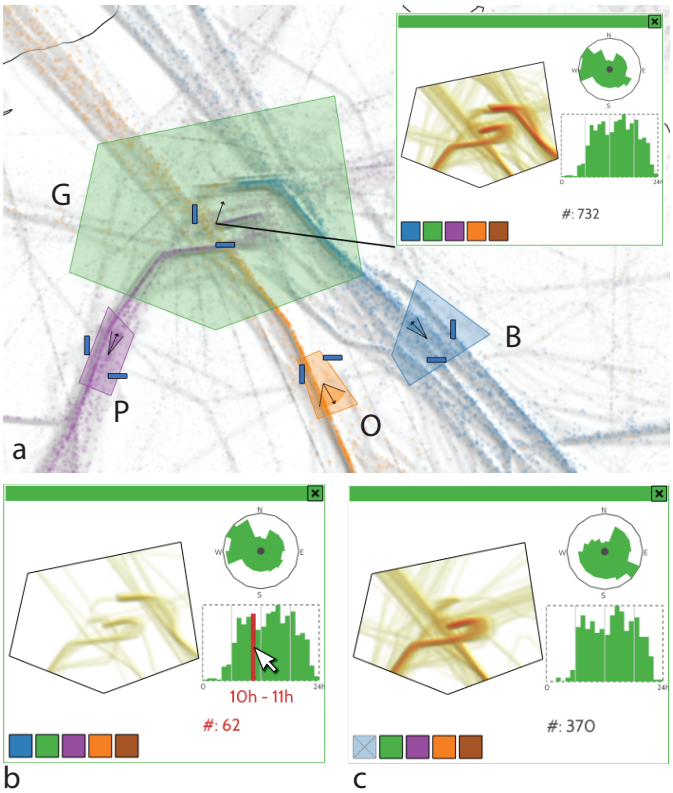

Fig. 6. Three different sets of traffic flows have been selected: Purple (P), orange (O), and blue (B). Also, a green (G) area has been created that does not select any traffic flow, but represents an area of interest. (a) A window has been opened showing the dynamics of the traffic in the green area, *e.g.*, distribution of traffic over time, and the distribution of directions. (b) By hovering the mouse over the time histogram, different time instances can be explored. (c) The color buttons in the bottom of the window show which traffic flows are visualized in the window. Here, the blue traffic flow has been deselected.

7 Traffic Flow Analysis

The user should be enabled to investigate the dynamics of the selected traffic flows, *i.e.*, how the traffic flows evolve over time (R3). Specifically, the usage of an area over time by objects moving through the selected areas, and the change of the directions of these objects over time should be presented. Also, the user should be enabled to compare the dynamics of multiple selections (R3).

We realized this by showing windows on demand for any selection area, that serve as annotations to the selected areas. These windows show the number of trajectories within the selection area over time using a histogram, a density map showing the trajectories within the selected area, and a polar area diagram showing the distribution of directions similar to a *wind rose* plot [32]–see Figure 6a. By hovering the mouse over bars of the histograms, the user can investigate specific time intervals. Both the density map and the direction diagrams show the density and direction, respectively, of the highlighted time interval–see Figure 6b and the supplemental video. To aid comparisons, all windows are linked, that is, the same time interval is highlighted in all windows.

The windows visualize the information of all tracks visible within the selection areas. The user can deselect individual selection colors by clicking the color buttons at the bottom of the window–see Figure 6c. Selection areas may also serve as selections of areas of interest, instead of selections of traffic flows, for instance, to investigate the dynamics of selected traffic flows in a specific area. The green selection in Figure 6a is an example of such an area of interest selection, which does not select any traffic flows as can be seen by the direction range in its glyph, *i.e.*, the range is a line, which means the direction range is empty.

Traffic flows can be compared in multiple ways [24]. The user can juxtapose multiple windows, but the user can also superimpose the movable windows by dragging these on top of each other–see the supplemental video. The stack of floating windows is then displayed as a single window containing all density maps in the stack. Both the polar area diagrams and the histograms are stacked by, for each bin, depth sorting the values, from large (back) to small (front) similar to the braided graphs of Javed *et al.* [25]–see Figure 7a. We allow the user to choose between juxtaposition and superimposition because both approaches have their unique strengths [25]. While superimposition (space sharing) is better for comparing local maxima, juxtaposition is better for dispersed comparison.

Apart from stacking, the user can also investigate the difference between traffic flows by dragging a line between two selection areas while holding the shift key. A new floating window appears with density maps for the selected areas. In this window, the difference between the traffic flows is shown in both the direction diagrams and the histogram. In the direction diagrams we show, per direction bin, the absolute difference between the traffic flows by the color of the area using the color of the largest traffic flow. In the histograms we show, per time bin, the difference between the two traffic flows where bars are drawn above the line if the bottom traffic flow is larger, and below the line if the top traffic flow is larger–see Figure 7b.

8 Evaluation & Use Cases

In this section we discuss expert feedback and a number of use cases using a set of aircraft trajectories over France of a single day, and a data set of vessel trajectories near the Dutch coast. The aircraft data set contains $17,841$ flights with a total of $424,546$ sample points. The vessel data set contains $16,421$ vessels with a total of $420,335$ sample points.

Since we used a user centered design process, we conducted several evaluation sessions with air traffic controllers. This section reports the latest evaluation in which two air traffic controllers, that have not been involved in the design process, used our system and gave feedback. The two air traffic controllers have 10 years of experience in the Paris area: *Roissy Charles de Gaulle* and *Orly*, the two biggest airports in France. During the evaluation, we first presented the goal of the tools with its available features. Then we asked them to freely use the soft-

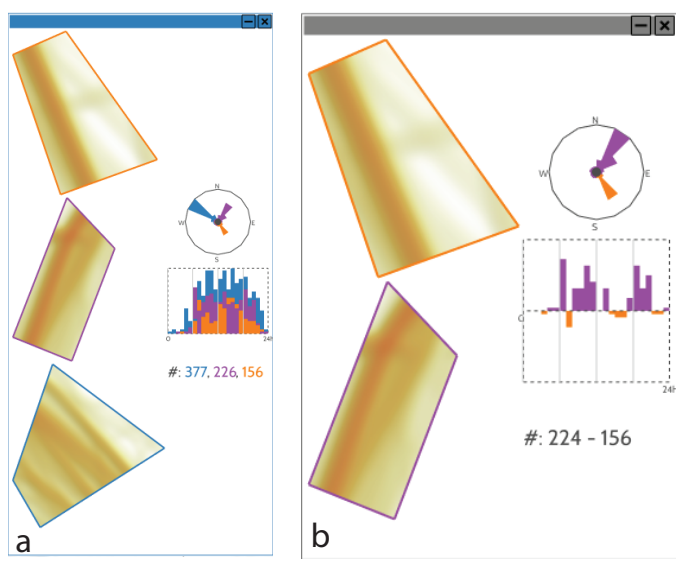

Fig. 7. (a) The user can compare multiple (in this case, three) traffic flows by *stacking* multiple windows, which results in stacked visualizations. (b) Additionally, the user can investigate the difference between two traffic flows.

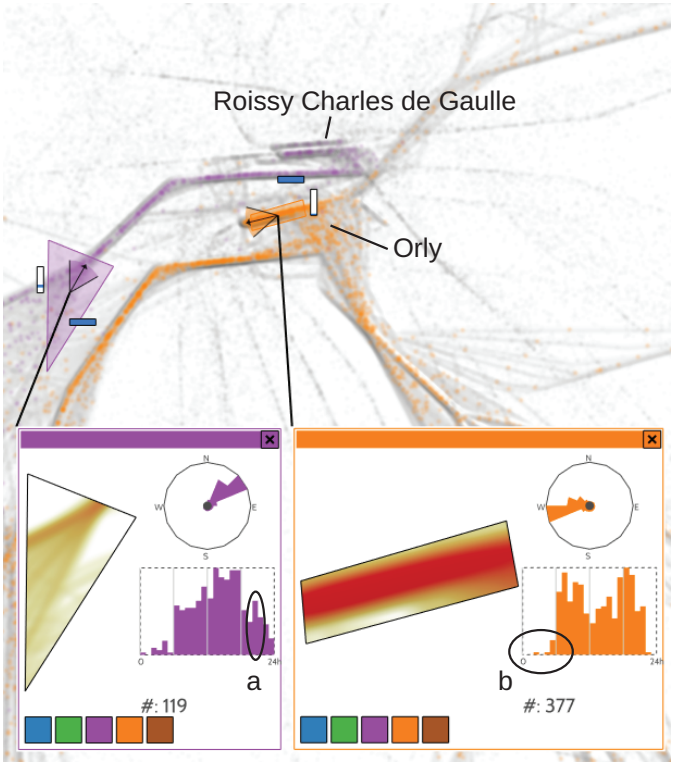

Fig. 8. Selections of a traffic flow coming into *Roissy Charles de Gaulle* from the south-west (purple) and a traffic flow landing at *Orly* (orange) with the outliers identified by the air traffic controllers: A peak of traffic around 8pm in the purple flow (a), and landing aircraft at Orly while the airport is not yet open (b).

ware while thinking aloud to better understand their reasoning.

The two air traffic controllers followed the same sequence of investigations: they first tried to validate what they already knew and then, they tried to understand the rationale of a number of outliers. They both first investigated the traffic flows of the airport they were most familiar with. What they found out about these traffic flows was consistent with what they already knew apart from two outliers. They investigated these outliers: unexpected traffic at 5am at Orly and a peak at 8pm at Roissy–see Figure 8. At 5am, Orly airport is closed, but a number of aircraft appear to have landed anyway. After some investigation, these aircraft appear to have landed at *Villacoublay* airport, a military air field very close to Orly. These aircraft are most likely medical evacuation services. Regarding the 8pm peak of traffic at Roissy Charles de Gaulle, the air traffic controller figured out that this corresponds to numerous Air France aircraft which fly back to Paris to spend the night at the airport before leaving again on the next morning.

Both controllers saw the main strengths of our tool in an educational setting, more specifically for air traffic controllers and analysts in training. To the best of their knowledge, no previous tool is available to display the recorded traffic flow with their directions. They reported that this tool provides a perfect visualization system to better understand the structure of the air space and how traffic flows *tangle*. Furthermore, the flow evolution is of a great interest to better grasp the temporal traffic density. Additionally, they said this tool can also be used for communication purposes, to show to a general audience how traffic is distributed over France. The displayed density map is really helpful to correctly grasp the actual flow density. Combined with the particle density, it creates a background image which better emphasize the local density.

The experts also saw many more operational usages: This tool can be complementary to existing ones to perform statistics. This can be a great asset in order to modify, forecast and assess air space structure and evolution. The few available tools do not display visual information but only textual and graphical statistics.

Our air traffic experts were regularly consulted during the design of our tool and the effectiveness of our design choices were regularly validated with the experts. They especially appreciated our selection widget to filter flows. The multidimensional filtering technique (rotation to a different data dimension) was also qualified as appealing to better understand how trajectories tangle in 3D.

We have chosen not to overlay our visualization of the aircraft data on top of a map because our domain experts know the region very well and consider a map unnecessary clutter. If our visualization is to be used for educational or communication purposes, however, a map would be advisable. In the vessel use case of Section 8.5, our visualization has been overlaid on top of a map.

In the following sections, we describe more use cases for our tool in both the air traffic domain and the Maritime domain.

8.1 Comparing Flows

Aircraft follow flight routes, ordered sequences of geographically referenced locations. Actual aircraft trajectories do not always follow the exact location of the flight route. Air traffic controllers can dynamically alter the route for reasons of safety and optimization. Investigation of such flight routes, or, traffic flows, is valuable to better understand airspace congestions and to improve flight regulation and safety. In this use case, we investigate two traffic flows crossing France in opposite directions. These flows correspond to flight transits between cities from the south-west to cities in the north-east, and vice versa: *Geneva*, *Lyon*, *Toulouse*, and *Madrid*. To avoid colliding aircraft, these parallel flows are geographically separated. Using our approach, we can select each flow with our selection widget, and define the altitude range corresponding to the desired flight routes–see Figure 1 where the altitude ranges are visualized using the small blue bars. The dynamics of these flows can be investigated separately, or can be directly compared by stacking their windows. As we can see, the traffic flow heading south-west is denser in the morning, while the traffic flow heading north-east is denser in the afternoon. According to our experts, this can be explained by passengers preferring to arrive in the morning in the south of Europe and wanting to return at the end of the day.

8.2 Landing and Take-off

In this scenario, we investigate landing and takeoff events at *Roissy Charles de Gaulle*, the main airport in France–see Figure 9a. To do so, we create two different selection boxes located before and after the runways. Next, we rotate to the altitude view to display the altitude and change the altitude range of the selections to only select aircraft that are landing or taking off, respectively, *i.e.*, aircraft with low altitude. In Figure 9a, blue traffic flows correspond to take-offs, while green traffic flows to landings. Both traffic flows are similar with around 850 aircraft each. The visualizations of the traffic flow dynamics show the specific departure and arrival sequences. Our experts explained the patterns as follows: In the time line of the departing aircraft (blue in Figure 9a), we can see (1) postal aircraft taking off between 0am and 5am, (2) two big departure sequences, called hubs, at 10am and 12am. In the time line of the landing aircraft (green in Figure 9a) we can see (3) several peaks in landing aircraft starting after 5am. The aircraft landing in the evening stay at the airport during the night. The peaks in landing and take-off generally do not coincide except for a peak between 10am and 11am (4). This information can be used to optimize aircraft scheduling.

8.3 Altitude

In this scenario, we investigate the aircraft distribution over different altitudes. In order to optimize fuel consumption, aircraft remain at the same altitude, also called Flight Level (FL) as much as possible. FLs are expressed in feet; for instance an aircraft flying at 30,000 feet (approximately 10 km high), has a Flight Level of 300. Stabilized aircraft heading east (with a direction between 0 and 179 degrees) have an *odd*

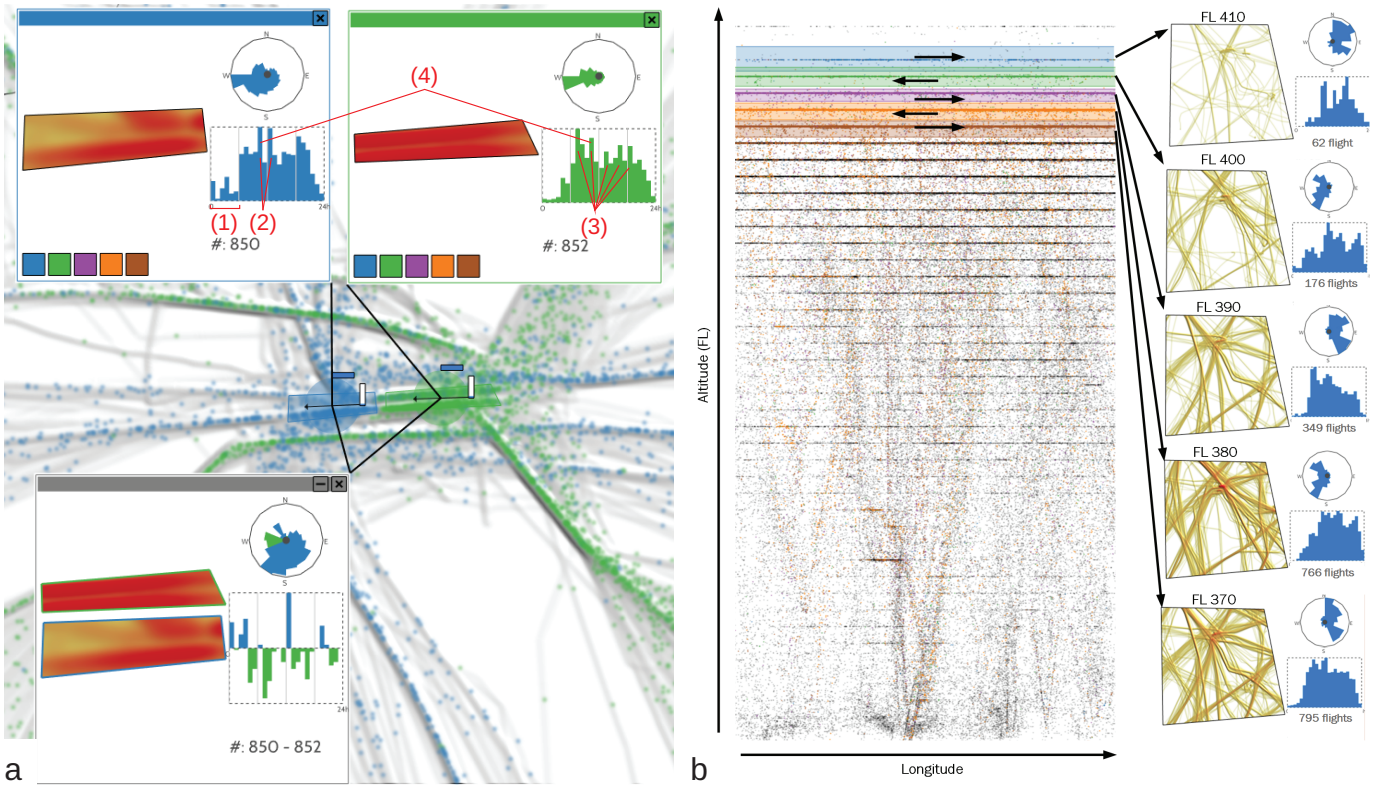

Fig. 9. (a) Aircraft traffic flows over the airport of *Roissy Charles de Gaulle*, France. Takeoff (blue) and landing (green) sequences have been selected and their dynamics can be compared using the windows. According to our experts we can see postal aircraft taking off (1), several large departure (2) and landing (3) sequences, and a peak in both landing and take-off at the same time. (b) The view has been rotated to aircraft altitude where we can see aircraft flying east and aircraft flying west fly at alternating flight levels.

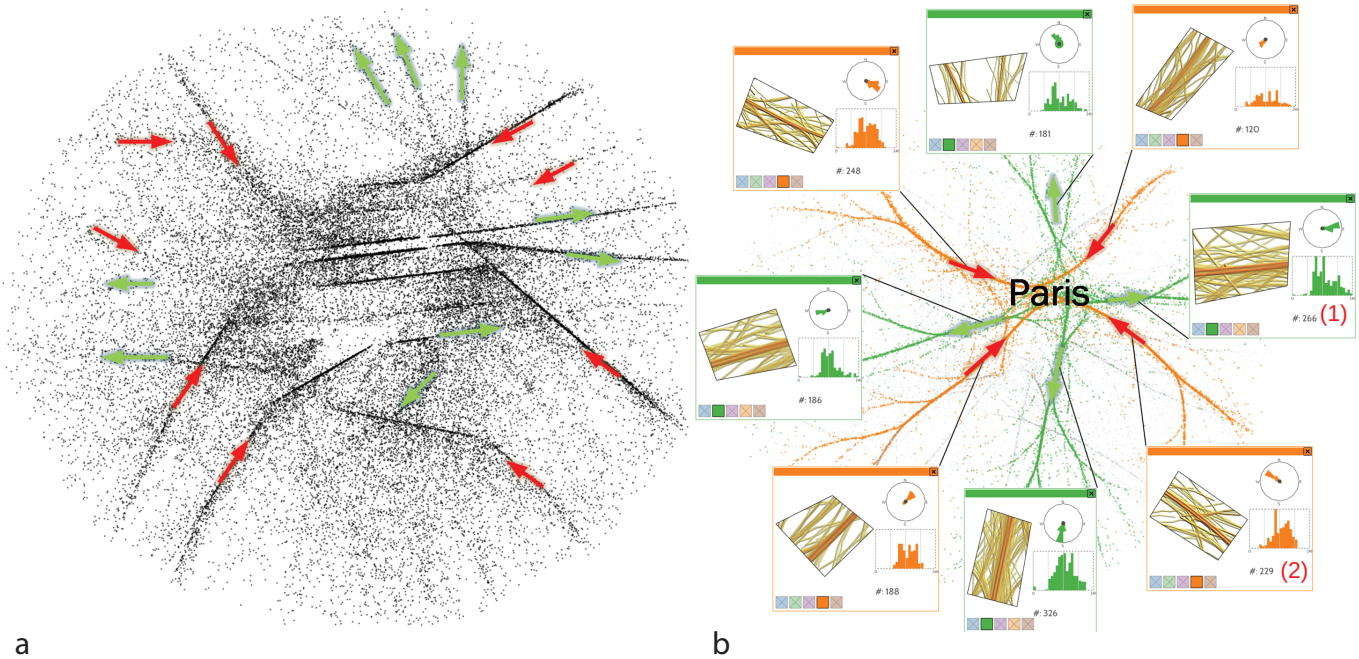

Fig. 10. (a) An overview of traffic flows over the Paris area. Outgoing traffic flows have been marked with the green arrows, while incoming traffic flows have been marked with a red arrow. (b) The traffic flows have been bundled, selected, and the dynamics of these traffic flows are displayed using the movable windows.

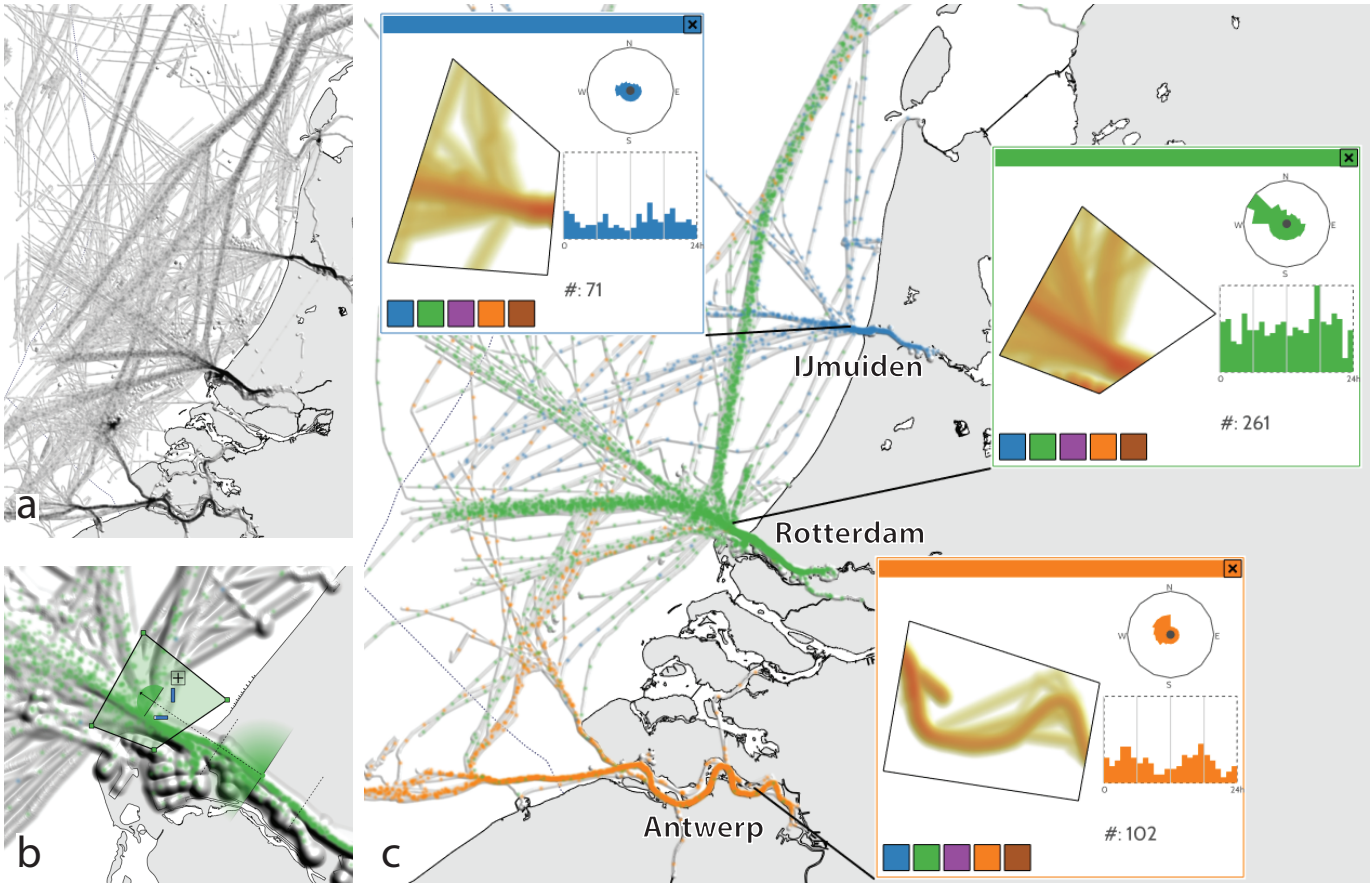

Fig. 11. (a) An overview of vessel traffic flows near the Dutch coast. (b) Vessels leaving the port of Rotterdam have been selected. (c) A visualization of the outgoing traffic of three harbors along the Dutch and Belgian coast: IJmuiden (blue), Rotterdam (green), and Antwerp (orange). The wave pattern in the histogram of the port of Antwerp is caused by the access to the port, the river *Scheldt*, being subject to the tides.

FL (i.e. 310, 330, 350, 370, etc.). Aircraft heading west (with a direction between 180 and 359 degrees) have an *even* FL (i.e. 300, 320, 340, 360, etc.). This mandatory rule helps to better separate aircraft traffic flows. When selecting flows by altitude, we can see the opposite aircraft direction between odd and even FL–see Figure 9b. Since only powerful aircraft can reach high altitudes, the flow density decreases with higher FL.

8.4 Traffic over Paris

In this use case, we investigate the traffic flow over *Roissy Charles de Gaulle*. When zooming into the Paris area, the flows appear entangled and spread out (see Figure 10a). As we have shown in other use cases, specifically in Section 8.2, this is no issue for our approach. A common approach, however, to deal with such clutter is to apply a visual simplification technique such as edge bundling [20]. Since flows are oriented, we use an extended version of the edge bundling technique which bundles trajectories with compatible directions [21]. This technique reduces visual clutter by aggregating edges into bundled flows. Edge Bundling provides a trade-off between empty spaces and overdrawing [19, 21]. As a drawback, the trajectories are distorted and thus not geographically accurate compared to the original trails. As shown in Figure 10b, we can still easily select the desired flows using our approach. Figure 10b shows the double cross flow system (four incoming flows and four outgoing), which can be investigated and compared. We can see some flows operate more in the morning, such as flow (1) heading east, while other flows operate mainly after midday, such as flow (2) coming from the south-east. Peaks correspond to hubs, *i.e.*, when more aircraft arrive at the same time to maximize efficiency.

8.5 Harbor Infographic

In this use case we use vessel trajectory data of a single day near the Dutch coast based on AIS [23] tracks–see Figure 11a. We show how our approach can be used to quickly construct an infographic-like visualization to compare the outgoing traffic of three harbors along the Dutch and Belgian coast: *IJmuiden*, *Rotterdam*, and *Antwerp*–see Figure 11c and the supplemental video. For each of the three harbors, we use a different selection color to select an area around the harbor mouth using a heading range of vessels moving away from the harbor–see Figure 11b. We increase the minimum of the velocity range so that we only select moving vessels, and open a window for each selection. By selecting the "render only selected trajectories" option, hiding unselected particles, and hiding the selection widgets, we get the visualization of Figure 11c.

Due to the colored particles, we can now see where vessels leaving the respective harbors typically go. The floating windows serve here as annotations for each harbor that show the dynamics of traffic leaving their respective harbors. We can see that the port of Rotterdam is by far the largest of the three. Also, we can see a wave pattern in the histogram of the port of Antwerp. This is because the access to the port of Antwerp, the river *Scheldt*, is subject to the tides.

9 Conclusions & Future Work

In this paper, we explored the design space to visualize and interactively explore traffic flows in moving object data sets. Traffic flows have intrinsic properties that can be displayed and analyzed even in dense visualizations: location, direction, and intensity. In addition, traffic flows can have other dimensions, such as altitude, speed, type of moving object, etc., which can evolve over time. The user can discover and identify traffic flows using a visualization which is a combi-

nation of a density map and a particle system. Then, using our novel selection widget, these traffic flows can be selected. When selected, the user can explore traffic flow dynamics using annotation windows, which can be dragged on top of each other to compare multiple traffic flows. We demonstrated our work using a number of use cases, which have been validated by air traffic analysts.

As future development, we plan to extend this software together with air traffic control practitioners to provide "a what if" system that enables the user to remove, change, or simulate trajectories. This will help the user to understand the impact of traffic flow modifications: What if this traffic flow is redirected in that direction? What is the impact on the other traffic flows? This can, however, also be used for other data sets. For example, to study what the effect on shipping lane usage is if more vessels leave the harbor in the morning.

We would also like to investigate the scalability of our approach by using larger data sets. Currently, we estimate parameters for the particle system, such as particle spacing and particle weight P_α, automatically for the data sets discussed in the use cases. These parameters are based on the average density of the data set. We would like to generalize this estimation so that we can automatically generate suitable parameters for any input data set. Additionally, we would like to improve this parameter estimation to take into account other relevant data characteristics such as the spread of the trajectories.

And finally, we would like to investigate generating infographic-style visualizations further using our approach. For example, the arrows shown in Figure 10b and similar annotations can be generated automatically, but currently are not.

Acknowledgments

We would like to thank the air traffic controllers and domain experts for their expert feedback during the design process. This publication was supported by the Dutch national program COMMIT. The research work was carried out as part of the Metis project under the responsibility of Embedded Systems Innovation by TNO with Thales Nederland B.V. as the carrying industrial partner.

References

[1] G. Andrienko and N. Andrienko. Visual exploration of the spatial distribution of temporal behaviors. In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 799–806, July 2005.

[2] G. Andrienko and N. Andrienko. Spatio-temporal aggregation for visual analysis of movements. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on*, pages 51–58, Oct 2008.

[3] G. Andrienko, N. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel. Scalable analysis of movement data for extracting and exploring significant places. *IEEE Trans. Vis. Comput. Graphics*, 19(7):1078–1094, July 2013.

[4] N. Andrienko and G. Andrienko. Designing visual analytics methods for massive collection of movement data. *Cartographica*, 42:117138, 2007.

[5] N. Andrienko and G. Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, 17:205 –219, feb. 2011.

[6] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A review of temporal data visualizations based on space-time cube operations. In *Eurographics State of the Art Reports*, EG STARs. Eurographics Association, 2014.

[7] J. Blaas, C. Botha, E. Grundy, M. Jones, R. Laramee, and F. Post. Smooth graphs for visual exploration of higher-order state transitions. *IEEE Trans. Vis. Comput. Graphics*, 15(6):969–976, Nov 2009.

[8] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Trans. Vis. Comput. Graphics*, 19(12):2376–2385, Dec 2013.

[9] K. Buchin, B. Speckmann, and K. Verbeek. Flow map layout via spiral trees. *IEEE Trans. Vis. Comput. Graphics*, 17(12):2536–2544, Dec 2011.

[10] S. Buschmann, M. Trapp, and J. Dollner. Real-time animated visualization of massive air-traffic trajectories. In *Cyberworlds (CW), 2014 International Conference on*, pages 174–181, Oct 2014.

[11] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, Jan. 2009.

[12] S. Dodge, R. Weibel, and A.-K. Lautenschtz. Towards a taxonomy of movement patterns. *Information Visualization*, 7(3-4):240–252, 2008.

[13] N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Trans. Vis. Comput. Graphics*, 14(6):1539–1148, Nov 2008.

[14] N. Ferreira, J. Poco, H. Vo, J. Freire, and C. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Trans. Vis. Comput. Graphics*, 19(12):2149–2158, Dec 2013.

[15] D. Guo and X. Zhu. Origin-destination flow data smoothing and mapping. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2043–2052, Dec 2014.

[16] M. Harrower and C. A. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[17] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 127–130, 2002.

[18] C. Hurter, R. Alligier, D. Gianazza, S. Puechmorel, G. Andrienko, and N. Andrienko. Wind parameters extraction from aircraft trajectories. *Computers, Environment and Urban Systems*, 47(0):28 – 43, 2014. Progress in Movement Analysis Experiences with Real Data.

[19] C. Hurter, O. Ersoy, S. Fabrikant, T. Klein, and A. Telea. Bundled visualization of dynamicgraph and trail data. *IEEE Trans. Vis. Comput. Graphics*, 20(8):1141–1157, Aug 2014.

[20] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. *Computer Graphics Forum*, 31(3pt1):865–874, 2012.

[21] C. Hurter, O. Ersoy, and A. Telea. Smooth bundling of large streaming and sequence graphs. In *Visualization Symposium (PacificVis), 2013 IEEE Pacific*, pages 41–48, Feb 2013.

[22] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: Spreading aircraft trajectories across views to support iterative queries. *IEEE Trans. Vis. Comput. Graphics*, 15(6):1017–1024, Nov 2009.

[23] ITU. Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile band. *Recommendation ITU-R M.1371-4*, 2010.

[24] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE*, pages 1–8. IEEE, 2012.

[25] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, Nov 2010.

[26] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 97–104, 2001.

[27] R. Krüger, D. Thom, M. Wrner, H. Bosch, and T. Ertl. TrajectoryLenses a set-based filtering and exploration technique for long-term trajectory data. *Computer Graphics Forum*, 32(3pt4):451–460, 2013.

[28] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum*, 29(6):1807–1829, 2010.

[29] P. Muigg, M. Hadwiger, H. Doleisch, and E. Gröller. Visual coherence for large-scale line-plot visualizations. *Computer Graphics Forum*, 30(3):643–652, 2011.

[30] R. Scheepens, N. Willems, H. van de Wetering, G. Andrienko, N. Andrienko, and J. J. van Wijk. Composite density maps for multivariate trajectories. *IEEE Trans. Vis. Comput. Graphics*, 17:2518 –2527, dec. 2011.

[31] R. Scheepens, N. Willems, H. van de Wetering, and J. J. van Wijk. Interactive density maps for moving objects. *IEEE Comput. Graph. Appl.*, 32:56 –66, jan.-feb. 2012.

[32] D. H. Slade. Meteorology and atomic energy, 1968. Technical report, Environmental Science Services Administration, Silver Spring, Md. Air Resources Labs., 1968.

[33] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko. Stacking-based visualization of trajectory attribute data. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2565–2574, Dec 2012.

[34] S. van den Elzen and J. J. van Wijk. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2310–2319, Dec 2014.

[35] N. Willems, H. van de Wetering, and J. J. van Wijk. Visualization of vessel movements. *Computer Graphics Forum*, 28(3):959–966, 2009.

[36] J. Wood, A. Slingsby, and J. Dykes. Visualizing the dynamics of London's bicycle-hire scheme. *Cartographica*, 46(4):239–251, 2011.