Clustering Trajectories by Relevant Parts for Air Traffic Analysis

Gennady Andrienko, Natalia Andrienko, Georg Fuchs, and Jose Manuel Cordero Garcia

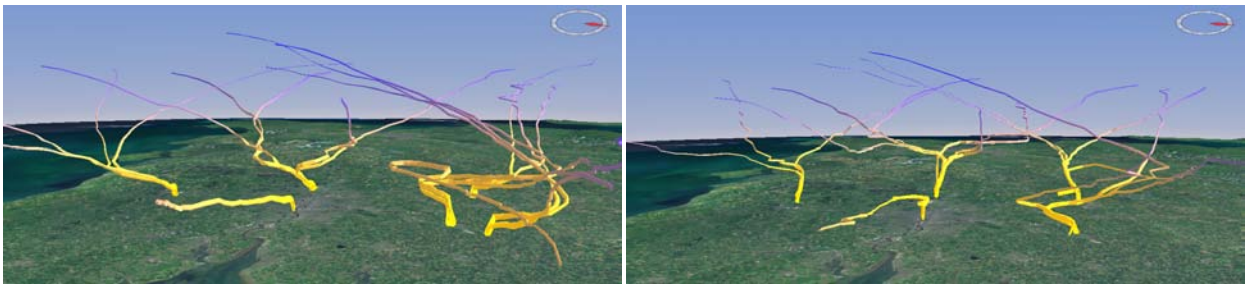

Fig. 1. Different approach routes to London airports are represented in a 3D view by central trajectories of density-based clusters of relevant parts of flight trajectories. The images show the routes that were used in two days with different wind parameters.

Abstract—Clustering of trajectories of moving objects by similarity is an important technique in movement analysis. Existing distance functions assess the similarity between trajectories based on properties of the trajectory points or segments. The properties may include the spatial positions, times, and thematic attributes. There may be a need to focus the analysis on certain parts of trajectories, i.e., points and segments that have particular properties. According to the analysis focus, the analyst may need to cluster trajectories by similarity of their relevant parts only. Throughout the analysis process, the focus may change, and different parts of trajectories may become relevant. We propose an analytical workflow in which interactive filtering tools are used to attach relevance flags to elements of trajectories, clustering is done using a distance function that ignores irrelevant elements, and the resulting clusters are summarized for further analysis. We demonstrate how this workflow can be useful for different analysis tasks in three case studies with real data from the domain of air traffic. We propose a suite of generic techniques and visualization guidelines to support movement data analysis by means of relevance-aware trajectory clustering.

1 Introduction

Movement data are widely collected nowadays in various domains, such as animal ecology, sports, and land, sea, and air traffic. The increasing availability of air traffic data through live trackers such as FlightRadar24 [3] and FlightAware [2] raised substantial interest of general public and provoked numerous artistic visualizations and animations of air traffic in the web [38] and mass media (e.g. [1]).

In air traffic management and control, movement data describing planned and actual flights are used for planning and monitoring purposes with the goal of increased utilization of air space capacities without compromising the safety of passengers and cargo, nor timeliness of flights. The significant increase of performed flights in the limited air space multiplies the complexity of planning, management, and monitoring tasks, which is accompanied by continuously growing amounts of data that need to be analyzed. There is a need for powerful analysis methods suitable for the data and tasks in the aviation domain. Our work has been motivated by tasks requiring the analysis of flight routes. We propose an analytical workflow and a suite of computational, visual, and interactive techniques, the core of which is density-based clustering of trajectories by the similarity of the followed routes. Despite the original motivation, the workflow and techniques are generic and can be used for movement analysis in various domains.

Clustering is an efficient and commonly used instrument for handling large amounts of complex data and creating understandable overviews of properties and patterns that exist in the data. Clustering of trajectories, which are complex spatio-temporal constructs, requires specific distance functions to assess the (dis)similarity between the trajectories.

There exist analysis tasks for which only certain parts of trajectories are relevant. The analysis needs to be focused on these relevant parts while keeping the integrity of the available trajectories. For example, for different analysis tasks in the aviation domain, it may be necessary to focus on the initial or final parts of the flights (to analyze takeoff or landing schemes), or to ignore these parts and consider the variety of the routes from the origins to the destinations, or to deal with the parts of the flights within a certain area or volume in the air space. When clustering is used for such tasks, it needs to be applied only to the task-relevant parts of the trajectories. A straightforward approach is to extract the relevant parts from the trajectories and supply them to the clustering algorithm. However, the division into relevant and irrelevant parts may be temporary and change throughout the analysis process. It may be necessary to cluster trajectories based on different selections of relevant parts while the integrity of the trajectories needs to be preserved. Hence, clustering of trajectories needs to be implemented so that the current selection of task-relevant parts is taken into account.

Relevance-aware clustering of trajectories is the topic of this paper. Based on the available related work (Sect. 3), we propose (1) visualization guidelines for supporting interactive selection of task-relevant parts of trajectories (Sect. 4.1); (2) an approach to relevance-aware density-based clustering of trajectories (Sect. 4.2); (3) a method for summarized representation of clusters by their central trajectories (Sect. 4.3); (4) demonstration of the effectiveness of the proposed techniques through case studies of real-world air traffic management problems (Sect. 5).

The main contribution of the paper is a set of techniques and visualization guidelines for supporting relevance-aware cluster analysis of trajectories (**RACAT**). We begin with a brief introduction of the workflow comprising this kind of analysis, which will help us to clearly position our work among the related works discussed in Sect. 3.

- *Gennady Andrienko and Natalia Andrienko are with Fraunhofer IAIS and City University London. E-mail: gennady.andrienko@iais.fraunhofer.de.*
- *Georg Fuchs is with Fraunhofer Institute IAIS. E-mail: georg.fuchs@iais.fraunhofer.de.*
- *Jose Manuel Cordero Garcia is with CRIDA (Reference Center for Research, Development and Innovation in ATM). E-mail: jmcordero@e-crida.enaire.es*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

2 RACAT WORKFLOW IN A NUTSHELL

The input to the workflow is a set of trajectories $Tr = \{tr_1, tr_2, ..., tr_k\}$, where each trajectory is a sequence of points $tr_i = < p^i_1, p^i_2, ..., p^i_{L(i)} >$, $L(i)$ being the length of the trajectory tr_i. Each point p^i_j describes a spatial location and, possibly, movement attributes of a moving object at some time moment or during an interval; that is, a point is a tuple $< t, l, \{a\} >$, where t is a time reference, l is a location in a two- or three-dimensional space, and $\{a\}$ is a combination of values of movement attributes. At the overall level, the workflow consists of three steps:

Filter: By interactive visually supported filtering, select relevant parts of the trajectories. The filtering is applied to the *points* of the trajectories. Filter conditions may refer to any of the components $< t, l, a >$, i.e., points can be filtered based on the times, spatial locations, and/or movement attributes. The output of this step is a set of *relevance masks* $M(Tr) = \{m_1, m_2, ..., m_k\}$. A mask m_i referring to trajectory tr_i is a sequence of relevance flags $m_i = < r^i_1, r^i_2, ..., r^i_{L(i)} >$; $r^i_j \in \{1, 0\}$. A point p^i_j is called *active* when $r^i_j = 1$ and *inactive* otherwise. The filter conditions defining the masks need to be set so that the active points correspond to relevant trajectory parts as conceived by the analyst.

Cluster: Apply a generic density-based clustering algorithm to the set of trajectories Tr with their masks $M(Tr)$ using a trajectory-specific distance function $D(tr_1, m_1, tr_2, m_2) \rightarrow R_{\geq 0}$ that takes into account the relevance masks. For a given pair of trajectories tr_1 and tr_2 with the respective masks m_1 and m_2, the function returns a non-negative real number that is considered as the distance between the trajectories tr_1 and tr_2. On this basis, the clustering algorithm defines groups of close trajectories, called *clusters*, and marks trajectories that are not close enough to others as noise. The output of this step is an assignment of cluster labels to the trajectories: $L(Tr) = \{L_1, L_2, ..., L_k\}$, where $L_i \in \{C_1, C_2, ..., C_m, noise\}$, $C_1, C_2, ..., C_m$ being the clusters.

Summarize and analyze: Create summary representations $S_1, S_2, ..., S_m$ of the clusters $C_1, C_2, ..., C_m$ and apply visualization and analysis to the set $\{S_1, S_2, ..., S_m\}$.

The RACAT workflow is performed in an iterative manner. In the filtering step, the relevance masks $M(Tr)$ need to be evaluated by the analyst: Do they correctly distinguish the relevant and irrelevant parts of the trajectories? This requires visualization support enabling the analyst to see both the active and inactive trajectory parts and clearly differentiate them. Based on the visual evaluation, the analyst may need to change the filtering conditions and evaluate the new result. The second step involves evaluation of the clustering outcomes: Are the clusters interpretable and internally coherent while the proportion of the noise is not too high? The clustering tool may need to be run several times with different parameter settings until the result is good enough. The evaluation is supported by visualization of $\{C_1, C_2, ..., C_m, noise\}$ as well as by numeric measures of the cluster quality, such as statistics of distances between the members of each C_i. After summarizing the clusters and analyzing $\{S_1, S_2, ..., S_m\}$, the analyst may need to apply the procedure to other parts of trajectories, which implies returning to the filtering step and changing the filter conditions.

In the next section, we discuss the works related to each step of the workflow. As we are going to demonstrate the use of RACAT for analysis tasks in air traffic management, we also include a discussion of the existing visual analytics works dealing with air traffic data.

3 RELATED WORK

3.1 Interactive visualization and filtering of trajectories

Visualization of multiple trajectories faces the problem of overplotting. Decreasing opacity levels can reveal some patterns (mainly the density variation) but many important details remain hidden. Schematic representations [33] and edge bundling [23, 37] partly solve the problem but introduce undesired displacements. Trajectories in 3D space require explicit representation of altitudes or depths, e.g., by per-segment colors [3] or by using 3D displays [20]. Movement attributes along a route can be represented by different geometry types, colors, and glyphs [52], but such visualizations increase display clutter and thus require interactive selection of trajectories to explore. Interactive selection that temporarily hides a part of data is often called *filtering*.

As spatio-temporal objects, trajectories can be filtered based on the spatial and/or temporal properties of the whole trajectories or their constituent points [10]. There are two common approaches to interactive filtering of time-referenced data: by selecting a continuous time interval [7] or according to the positions of the time references within a time cycle [28, 29]; both can be applied to trajectories. Trajectories can also be filtered based on their attributes: qualitative, such as object category (e.g. airline or aircraft type), or quantitative, such as duration, path length, or maximal speed [10]. The FromDaDy system [34] enables selecting trajectories by sketching shapes of interest on a 2D map or in a time graph depicting e.g., altitude dynamics. Selection operations can be applied sequentially to previously selected subsets.

The RACAT workflow involves filtering that selects parts of trajectories rather than complete trajectories. Filter conditions can be specified in terms of attributes of the trajectory points. Paper [13] proposes a taxonomy of attributes that can be derived from the spatial positions alone or in combination with data describing the spatial, temporal, or spatio-temporal context of these positions. Filtering can also be done according to a selection of time intervals satisfying interactively specified query conditions [16], which can be set based on aggregated properties of the overall movement, occurrences of events, or values of any time-dependent attributes, such as weather parameters. Trajectory parts that occurred during the time intervals for which query conditions hold are treated as active and the remaining parts as inactive.

Positioning of our work. As explained in Sect. 2, evaluation of filtering results needs to be supported visually in such a way that active and inactive parts of trajectories can be seen and clearly differentiated. We propose (Sect. 4.1) several approaches addressing this requirement.

3.2 Trajectory clustering

A possible approach to trajectory clustering is to represent each trajectory by a set of features (attributes), and apply generic methods suitable for clustering of relational data. The features may include the path lengths, durations, average speeds, start and end times, or other characteristics expressed by numeric or qualitative attributes attached to whole trajectories. This approach is not suitable for tasks where the traveled routes are in focus. When all trajectories have the same lengths, the coordinates of all trajectory points can be used as independent features. There exists an example of representing weekly dynamics of traded stocks by equal-length trajectories in an abstract space spanned by two performance indicators, and applying SOM clustering to the coordinates from these trajectories treated as features [48]. However, equal-length trajectories rarely occur in real movement data.

Due to typically high variability among trajectories, it is useful to apply density-based clustering methods, such as DBScan [24] and Optics [17], that separate clusters of similar trajectories from noise. These methods are used in combination with special distance (dissimilarity) functions [36, 42], such as the Euclidean distance between simultaneously reached positions [41], Frechet distance [9], dynamic time warping (originally proposed for time series [18] but adapted to trajectories), and longest common sub-trajectory [50]. Pelekis et al. [42] give an overview of trajectory similarity research and describe several sophisticated distance measures. Other approaches include application of sequence mining methods to a symbolic representation of trajectories as sequences of predefined location labels [26] and a generative model approach that involves grouping of vector fields [25]. Sacha et al. [45] present an interactive visual environment in which the user can explore the impact of different similarity measures, albeit in combination with k-Means / k-Medoids clustering methods.

Progressive clustering [44] entails iterative application of clustering with different distance functions or parameter settings to subsets of data that are selected based on the results of previous clusterings. For example, a simple distance function is used to obtain initial clusters, from which the analyst selects particular clusters of interest and applies the next clustering with a more computationally-intensive function only to the members of these clusters. This approach not only reduces the computational complexity but also makes the resulting clusters easier to interpret by the user as compared to application of sophisticated dis-

tance measures that simultaneously account for multiple heterogeneous properties of trajectories. The idea of progressive clustering can be extended to very large sets of trajectories [14]: after building initial clusters from a sample of trajectories, they are inspected and, possibly, edited by the analyst, and then the remaining trajectories are classified according to their similarity to representative trajectories of the clusters.

Besides clustering methods dealing with complete trajectories, there are also methods for subtrajectory clustering [40]. Trajectories are previously divided into pieces that can be represented in a simplified manner, for example, by straight line segments; then clustering is applied to these simplified objects. The result is clusters of trajectory pieces but not clusters of the entire trajectories. Therefore, this kind of clustering can be used for tasks in which the integrity of original trajectories is not important, e.g., those that focus on the variation of the overall movement characteristics across a territory. Several approaches to clustering of consecutive trajectory segments according to their features are compared in paper [51]. Such clustering can be used for discriminating different parts of trajectories, e.g., flight phases.

Specialized methods for clustering of flight trajectories have been developed in the domain of air traffic research [22, 47]. However, these approaches do not provide sufficient flexibility for selecting parts of trajectories that are relevant to specific analysis tasks.

So far, filtering and clustering of trajectories have existed as separate topics in different research disciplines, visual analytics and data mining. Integration of database querying with clustering is technically supported in the M-Atlas system [27], which is built for supporting data mining workflows and uses visualization primarily for communication of final results. There is a need for a systematic approach that addresses all steps of the cluster analysis process, including visualization and interactive exploration of intermediate results that drive further analysis.

Positioning of our work. We propose an integrated approach that supports the entire RACAT workflow. This includes application of clustering to parts of trajectories selected by means of interactive filtering.

3.3 Summarization of clusters of trajectories

A trajectory cluster can be summarized by constructing a single representative trajectory, called *central trajectory* [49]. The general idea is to create groups of points taken from different trajectories, select or construct a representative point from each group, such as the mean or median position of the group, and arrange the representative points in a sequence according to the ordering of the points in the original trajectories. Kreveld et al. [49] show that the choice of the mean or median positions as representative points may sometimes lead to undesirable artifacts in the resulting central trajectory, such as segments crossing a lake while the original trajectories go around. They propose a sophisticated (and computationally demanding) method for selecting representative positions that minimizes such artifacts.

When trajectories represent simultaneous movement of a group of objects, the points from different trajectories can be grouped based on common time references [15, 49], which is not applicable when trajectories are not synchronous. Lee et al. [40] propose a time-agnostic method suitable for clusters of sub-trajectories represented by straight line segments. For these segments, the method builds the average direction vector, sweeps a perpendicular line along this direction, finds the intersections of this line with the segments, and takes the average coordinates of the intersection points as components of the central trajectory. This trajectory represents only a subset of segments from the original trajectories. The authors do not propose a way to merge these partial representations into full trajectories.

Positioning of our work. We propose a simple and efficient method for constructing central trajectories of clusters of geometrically similar trajectories, in which points from the trajectories are put in groups based on their spatial proximity irrespective of the time references.

3.4 Visual Analytics for air traffic analysis

Currently deployed and perspective software tools in the domain of air traffic management (ATM) support relatively simple queries and include rudimentary visualizations, such as maps showing individual movements and time histograms with aggregated flight data [32]. More advanced techniques are being proposed by visual analytics researchers. Methods for detection of holding loops, missed approaches, and other aviation-specific patterns were implemented in a system integrating a moving object database with a visual analytics environment [46]. Albrecht et al. [8] calculate air traffic density and, taking into account the aircraft separation constraints, assess the conflict probability and potentially underutilized air space. The traffic density and conflict probability are aggregated over different time scales to extract fluctuations and periodic air traffic patterns. Hurter et al. [31] propose a procedure for wind parameter extraction from the statistics of the speeds of planes that pass the same area at similar flight levels in different directions. Paper [19] describes techniques for studying the dynamics of landings at Zurich airport with the goal to detect cases of violating the rules that prohibit night-time landings from the north, which produce strong noise in populated regions. The detected violations can be examined in relation to weather conditions and air traffic intensity.

Positioning of our work. In close collaboration with aviation experts, we conducted several case studies addressing different practical problems from the ATM domain requiring the analysis of flight routes, in which the RACAT workflow was successfully applied.

4 Our approach

We present our approach in the following three subsections referring to the steps of the RACAT workflow introduced in Sect. 2.

4.1 Filtering

It is not our goal to propose specific techniques for interactive filtering. We assume that any combination of techniques enabling space-based, time-based, and attribute-based filtering of trajectory points and segments [10] can be used to select relevant parts and focus on the visual support to the evaluation of the resulting selection. The following requirements need to be fulfilled:

- **R1**: The analyst can see the active parts of the trajectories separately, to check whether they only include what is relevant.
- **R2**: The analyst can see the inactive parts separately, to check whether they do not include what is relevant.
- **R3**: The analyst can see the active parts in the context of the inactive ones, to have a more complete picture of the data and a reminder of the filter being in use.

Requirements R1 and R2 can be fulfilled by a visualization that hides inactive items in a combination with a tool for inverting the current filter conditions. To fulfill requirement R3, the active and inactive parts need to be shown together but represented in visually distinct ways. We propose several approaches to visual differentiation of active and inactive parts of trajectories shown in a map display or a 3D view:

- *filter-aware rendering*, which represents active and inactive parts using different line attributes. Thus, for inactive parts, it can use thinner or more transparent lines than for active parts, or represent active parts by solid lines and inactive by dashed or dotted lines;
- visualization of a *boolean attribute* that reflects the current filtering result, i.e., values 'true' and 'false' are attached to active and inactive points. The attribute needs to be dynamically updated each time when the filter conditions change. The values can be represented visually by distinct colors;
- *different level of detail* in showing active and inactive parts. Specifically, active parts can be shown in full detail while inactive can be aggregated, for example, in a density surface [39, 53].

Examples of using these approaches can be seen in Sect. 5. Filter-aware rendering is used in Fig. 3, where active parts of trajectories are represented by solid lines and inactive parts by dashed lines. A boolean attribute reflecting a filter is used in Fig. 8; the value 'true' is represented by blue and 'false' by red color. The visibility of either value can be interactively switched on and off. The visualization of active and inactive parts with different levels of detail is used in Fig. 9, where active parts are represented in detail by colored lines while inactive parts are aggregated in a density surface rendered using the "hill shading" technique [30]. In all examples, active and inactive parts and their spatial distributions can be clearly distinguished.

4.2 Clustering

The main requirement for the clustering step is that the clustering tool should directly use the result of the filtering, i.e., the set of masks $M(Tr)$, in combination with the original set of trajectories Tr; no prior data transformation should be necessary. The filter-aware clustering can be done using a generic clustering algorithm that permits application of external distance functions, particularly, special functions for trajectories [10, 44]. Thus, we use the density-based clustering algorithm Optics [17] with the distance function "route similarity" [10].

4.2.1 Filter-aware distance measurement

RACAT requires a distance function that assesses the similarity of two trajectories based on spatial, temporal, and/or thematic properties of their points or segments, rather than on overall properties of whole trajectories. The implementation of the chosen distance function needs to be modified so as to take the relevance mask into account, i.e., it needs to be a function $D(tr_1, m_1, tr_2, m_2)$ that measures the distance between tr_1 and tr_2 using only their active points selected by the masks m_1 and m_2, respectively. We have chosen and modified the distance function "route similarity", which scans a pair of trajectories from the beginning to the end, iteratively finds matching points, measures the distances between them, and computes the mean distance. A penalty is added for points that cannot be matched to sufficiently close points in the other trajectory. The penalty is computed as the sum of the distances from the unmatched points to the counterpart trajectories divided by the length of the matched parts.

Since filtering does not change trajectories but only defines relevance masks, it is possible to do clustering two or more times with different filter conditions. For comparison, the results can be visualized in several interactive maps allowing the clusters to be selected or hidden. The analyst can select one or a few clusters in one map; in response, this and other maps hide all trajectories except the cluster members. It is thus possible to see where the members of these cluster(s) appear in the outcomes of the other runs of clustering: whether they are in the noise, in a single cluster, or distributed over several clusters.

4.2.2 Progressive clustering

Progressive clustering [44] is a procedure in which clustering with different distance functions or parameter settings is applied to different subsets of data. It can be useful in exploring heterogeneous properties of trajectories by combining the capabilities of diverse distance functions [44]. It is also helpful when the data density greatly differs across the dataset, as, for example, flight frequencies along different routes.

Let us explain the problem of variant data density and how it can be solved by means of progressive clustering. Density-based clustering involves two parameters, the neighborhood radius *NR* and the minimal number of neighbors *NN* an object must have for being a core object of a cluster. These parameters determine the sizes and densities of clusters that will be extracted. With small *NR* and large *NN*, the algorithm may extract few very dense clusters and label the remaining objects as noise, thus missing important clusters with lower density. With large *NR* and small *NN*, the algorithm may produce very large clusters with high internal variance. It is reasonable to vary the parameter values depending on the densities in different data subsets, but the existing clustering algorithms apply the same settings everywhere. This limitation can be overcome in the following way. First, the densest clusters are extracted by using a small *NR* and large *NN*. Then the clustering tool is iteratively applied to the noise obtained from the previous run. In each step, the parameter settings are relaxed by increasing *NR* and/or decreasing *NN*. The process stops when no additional clusters can be obtained, or the obtained clusters are too small or too incoherent.

Progressive clustering can also be used for refinement of selected clusters that are internally incoherent. A result of clustering may include just a few "dirty" clusters (i.e., having high internal variance) while the other clusters are good enough. In such a case, it is reasonable to apply clustering with stricter parameter settings to just the members of the dirty clusters, preserving the good clusters. Based on our experience, we recommend to apply the next step of clustering to the dirty clusters together with the noise. This may not only subdivide the dirty clusters into smaller and cleaner clusters but may also attach some items from the former noise to the new clusters.

There is no general rules for choosing suitable values of the clustering parameters. Usually, it is necessary to make several clustering trials with different settings to understand how changes of *NR* and *NN* affect the outcomes and to find a combination that gives a good result, i.e., well separated, well interpretable, internally coherent clusters and a moderate proportion of noise. Reasonable initial values can be chosen by observing the spatial distribution of the data on a map, specifically, the distances between items that are visually perceived as clustered and the density of such item groups. Due to the necessity of multiple trials, visualization plays an essential role by enabling the analyst to visually assess the cluster quality. Additionally, the quality can be characterized based on the pairwise distances among cluster members. Statistics of the distances for the clusters can be computed and presented visually allowing the analyst to decide which clusters require refinement.

4.3 Summarization

Clustering results are often represented visually by coloring items representing individual cluster members according to their cluster membership, as in Fig. 9. To reduce the display clutter and gain a clearer representation of what is common between cluster members, it is desirable to represent clusters in a summarized form. The requirement to the summary representation is that it needs to show clearly the features based on which the trajectories have been clustered.

Specifically, if the clustering was done according to the similarity of the traveled routes, the cluster summaries need to capture and represent the common routes. This can be achieved by constructing the central trajectory of each cluster (CTC) [49]. As explained in Sect. 3.3, corresponding points from member trajectories need to be grouped, and the CTC is then made from representative points, which are usually the mean or median points of the groups. When trajectories describe simultaneous movement of multiple objects, point grouping is done based on the closeness of the points' time references. The point grouping task becomes more difficult when trajectories are not aligned in time. Thus, clusters built according to the similarity of traveled routes may consist of asynchronous trajectories. For building a CTC, points from the original trajectories need to be grouped based on their spatial proximity and ordering within the trajectories, regardless of the time references. We propose the following algorithm 1 for grouping points from trajectories that are assumed to follow similar routes.

The algorithm, which is schematically illustrated in Fig. 2, includes two operations, bisection and group refinement. At the beginning, two groups G_1 and G_2 are created from the start and end points of the trajectories and arranged in an ordered sequence $S = < G_1, G_2 >$. The bisection operation (lines 11-14) takes two consecutive point groups from S, and from each trajectory that has points in both groups, takes the midpoint between these two points, if such a point exists. The midpoints from multiple trajectories are put in a new group (G_3 in Fig. 2), which, in case of having points from at least a half of the set of trajectories, is inserted in S between the original two groups. The group refinement operation (lines 22-27) constructs the group center (average point) by computing the means or medians from the coordinates of the member points of the group. In Fig. 2, group centers are marked by star symbols. Then the refinement procedure tries to find in each trajectory between the points included in the previous and in the next groups of the sequence a not yet grouped point with the smallest distance to the center of the group being refined. If such a point exists, and it is closer to the group center than the originally taken point of this trajectory, the latter is replaced by the former. The refinement makes the groups more compact in space. After a group is refined, a new center is computed.

In Fig. 2, dashed ellipses encircle groups G_2 and G_3 before refinement. Point $p_{1.4}$ and $p_{2.3}$ were replaced in these groups respectively by points $p_{1.3}$ and $p_{2.4}$, which were closer to the group centers. The updated groups, denoted G'_2 and G'_3, are encircled by solid lines.

The computational complexity of algorithm 1 is $\mathcal{O}(N \times M)$, where N is the number of trajectories and M is the average number of points in one trajectory. The algorithm does not require the trajectories to have equal lengths, and it takes into account the current filtering, which

Algorithm 1 Point grouping for central trajectory construction

procedure GROUPPOINTSOFTRAJECTORIES(T)
  input: $T = \{t_1, t_2, ...t_N\}$ - set of trajectories;
  $t_i =< p_1^i, p_2^i, ..., p_{n_i}^i >$ - sequence of points of trajectory t_i
  $G_1 \leftarrow setOfStartingPoints(T)$
  $G_2 \leftarrow setOfEndingPoints(T)$
  $S \leftarrow < G_1, G_2 >$
  $refineGroup(G_1, 1, T)$ ▷ No check for G_{i-1}
  $refineGroup(G_2, 2, T)$ ▷ No check for G_{i+1}
  $i \leftarrow 1$
  repeat ▷ processing pair $< G_i, G_{i+1} >$ from S for bi
    $G_{middle} = \emptyset$
    for $j \leftarrow 1, N$ **do** ▷ for all traje
      if $\exists P_k^j \in G_i \wedge \exists P_m^j \in G_{i+1} \wedge k < m-1$ **then**
        $p \leftarrow middlePoint(P_k^j, P_m^j)$ ▷ bi
        $G_{middle}.insert(p)$
    if $\|G_{middle}\| \geq \|T\|/2$ **then**
      $S.insertAfter(i, G_{middle})$ ▷ inserted after $\|G_i\|$
      $refineGroup(G_{middle}, i+1, T)$ ▷ group refinement
    else
      $i \leftarrow i+1$
  until $i = S.length$
  return S ▷ ordered sequence of groups
procedure REFINEGROUP(G, i, T)
  $avgPoint \leftarrow averagePointFromSetOfPoints(G)$
  for $j \leftarrow 1, N$ **do** ▷ for all trajectories
    if $\exists p_q^j \in G \wedge \exists P_k^j \in G_{i-1} \wedge \exists P_m^j \in G_{i+1} \wedge k < m-1$ **then**
      $np \leftarrow findNearestPoint(avgPoint, t_{k+1}^j, t_{m-1}^j)$
      $G.replace(p_q^j, np)$

means that only active points of each trajectory t_i are included in the sequence $t_i =< p_1^i, p_2^i, ..., p_{n_i}^i >$ (line 2). After the algorithm terminates, the centers of the groups from S are taken in the order specified by S as the points of the CTC. Thus, in Fig. 2, the central trajectory is constructed from the centers of the groups G_1, G'_3, and G'_2 in this order. For movements in 3D space, the centers of point groups are constructed using all three coordinates of the member points, i.e., by taking the mean or the median of each coordinate.

CTCs are represented visually by lines on a map; three-dimensional CTCs can also be represented by tubes or ribbons in a 3D view, as in Fig. 1. The lines, tubes, or ribbons can be differently colored for distinguishing the clusters (this is not the case in Fig. 1, where the coloring represents the altitudes). Cluster cardinalities can be represented by proportional line widths, as in Figs. 4, 10, and 11. Some parts of a CTC may look zigzagged, which indicates the presence of substantial deviations of the original trajectories from a common route. A few CTCs with this feature can be seen in Fig. 10.

It may be useful to see CTCs in context of the original data, which need to be represented distinctly from the central trajectories. Thus, CTCs may be represented by thick solid lines and the original trajectories by thin and nearly transparent or dashed lines (Fig. 11-right), uniformly colored to reduce visual clutter, or they can be shown in a summarized form as a density surface (Figs. 10 and 11-left).

When points of trajectories are put in groups by algorithm 1, the points receive references to the groups, which are then replaced by references to the corresponding points in the resulting CTCs. These references can be used for creating statistical aggregates of any attributes associated with the original points. The aggregates are attached to the points of the CTCs and can be visually explored (e.g., as in Fig. 7) and used in further analyses. Attributes for points or segments of CTCs can also be derived from characteristics of their spatial context (Fig. 5).

It is worth noting that the use of CTCs is not limited to visual representation of clusters in a summarized form, but they can also be used in further analysis; in particular, computational operations can be applied to them. Thus, subsection 5.3 shows how CTCs are used for constructing a graph representing a network of traffic flows (Fig.11).

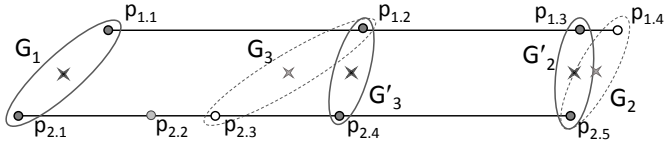

Fig. 2. A schematic illustration of Algorithm 1. The horizontal lines represent two trajectories, circles mark their points, and ellipses encircle point groups. Dashed lines encircle groups that were updated by refinement.

The techniques described in this section were evaluated by application to real-world data and tasks in the case studies presented below.

5 CASE STUDIES

As recommended by examples in the literature [35, 43], we used the expert review method for evaluating our work. All case studies were performed by two people, an expert in the domain of air traffic management (ATM) and an analyst. The former stated the overall analytical goals and posed more detailed task-related questions during the analysis process. The latter translated the tasks and questions into analytical operations, performed the operations using corresponding tools, presented visual displays of the results to the domain expert, and explained the representation thus helping the domain expert to interpret the results. In presenting the case studies, we shall describe the actions performed by the analyst and the judgments of the domain expert.

5.1 Revealing route choice criteria

There are many possible flight routes for each pair of origin and destination airports, and there are multiple criteria which can affect the route choices by flight operators (airlines) [21]. One of the criteria is the navigation charges, or taxes, that must be paid for crossing the airspaces of different countries. The charges are calculated based on the distances between the entry and exit points of the airspaces. The unit rates substantially differ among countries. Flight operators may strive to reduce flight costs by avoiding expensive airspaces, or minimizing the distances traveled across these airspaces. This strategy may compromise other aspects of a flight, such as the total path length, the cruise altitude and its constancy during the flight, and the probability of delays and rerouting due to high traffic density on some segments of the chosen route. The objective of this case study is to investigate how much the route choices may be affected by the differences in the charges and what may be the downsides of choosing cheap routes.

Route choices are analyzed based on flight plans created by operators. A flight plan includes a sequence of time-stamped way points, each having three coordinates: longitude, latitude, and flight level (altitude). From these points, the trajectory of the planned flight can be reconstructed. The data used for this study include a set of 1,717 flight plans (122,793 points) and a set of corresponding trajectories of the actual flights (171,134 points) from Paris to Istanbul performed during 5 months from January to May, 2016. This pair of cities was chosen for the study because regular flights between these cities were performed by 6 different flight operators, which could have different strategies in their route planning. The trajectories constructed from the flight plans are shown in Fig. 3, top. There is also a dataset specifying the boundaries of the changing zones in Europe and the unit rate for each. This information is shown in Fig. 3, top, by the background shading, where darker shades correspond to higher unit rates. The unit rate is the price in € per 100 kilometers of great-circle distance between zone entry and exit points for aircraft over 50 tons takeoff weight.

To achieve the analysis goal, it is necessary to identify the major routes that were used repeatedly rather than occasionally. These can be obtained by clustering the planned flight trajectories according to route similarity. However, trajectories that follow the same overall route may greatly differ in their departure and approach phases due to differences in active runways and operating procedures in effect, which mostly depend on weather conditions, and as a result may not be put in the same cluster. Hence, trajectory parts corresponding to departure and approach phases need to be excluded from clustering. For this purpose,

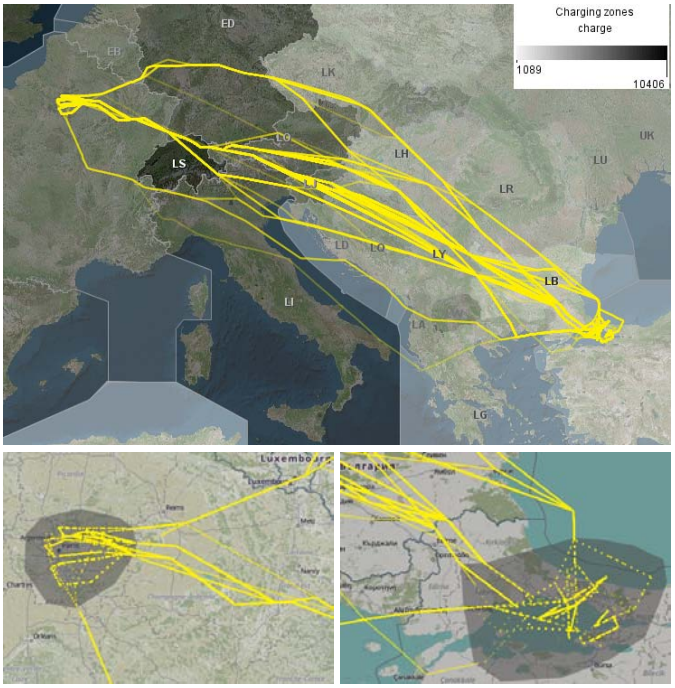

Fig. 3. Top: Trajectories of planned flights from Paris to Istanbul are shown on top of shading of the charging zones according to the unit rates. Bottom: The parts of the trajectories in the departure and arrival areas are irrelevant to the analysis.

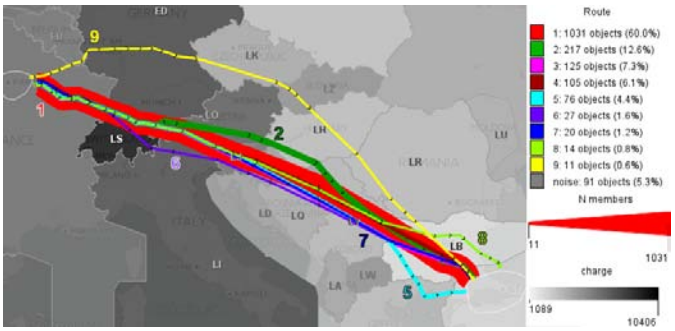

Fig. 4. The major flight routes extracted by clustering of the planned flight trajectories. The line widths are proportional to the cluster cardinalities.

the analyst roughly encircles the airport areas on the map and filters out the trajectory segments lying within these areas (Fig. 3, bottom).

By trying different clustering parameters, the analyst detects the existence of a very large cluster, which becomes highly inhomogeneous when a large neighborhood radius is chosen; however, with a small radius, too many trajectories go to noise. Therefore, the analyst applies progressive clustering. In the first step, the largest and densest cluster is extracted with the settings $NR = 3km$ and $NN = 10$. The next step of clustering with $NR = 6km$ and $NN = 10$ is applied to all trajectories except the members of the largest cluster. The result after both steps is 9 clusters with sizes not less than 10. Smaller clusters, which represent rarely used routes, are merged with the noise. As a result, the clusters include 1,626 trajectories (94.7%), from which 1,031 trajectories are the members of the largest cluster (cluster 1), and 91 trajectories (5.3%) are labeled as noise. The extracted routes are represented in Fig. 4 by the central trajectories of the clusters. The line widths are proportional to the cluster sizes, i.e., the number of times each route was chosen.

In Fig. 5, the charges (unit rates) along the routes are represented by proportional widths of line segments. While all but one routes cross the zone of Switzerland (labeled LS) with the highest charges, almost all of them travel a very small distance across this zone and only route 6 (violet) has a longer segment within the zone. After leaving zone LS, route 2 (green) deviates from the other routes and goes through zones LO (Austria) and LH (Hungary) avoiding more expensive zones LI

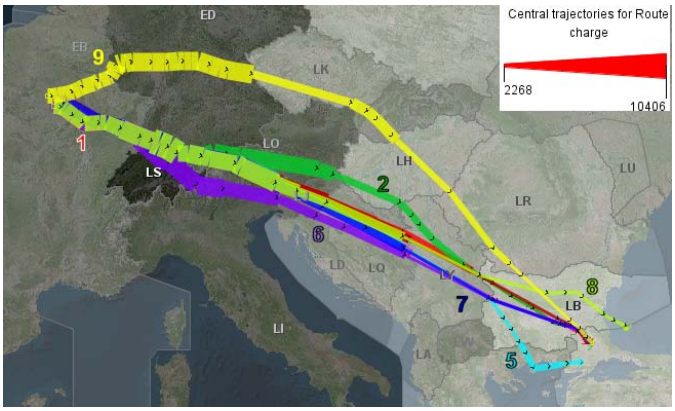

Fig. 5. The navigation charges per distance unit along the major flight routes are represented by proportional widths of line segments.

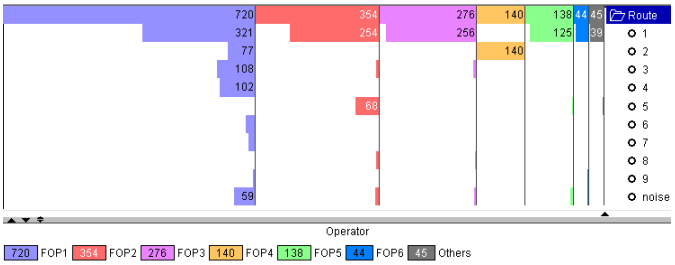

Fig. 6. The route choices of the major flight operators. Top most bar is flight totals across all routes, followed by route numbers.

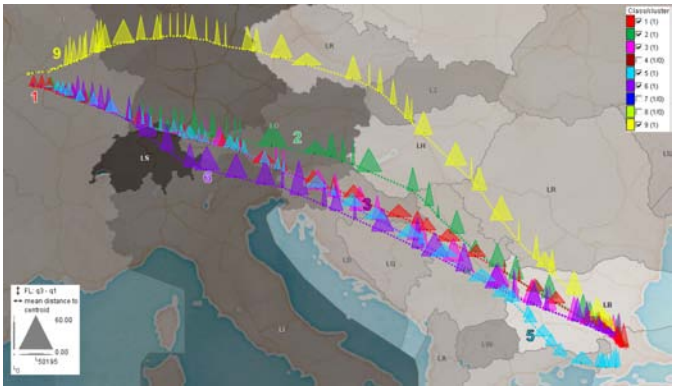

Fig. 7. The triangular symbols represent the flight position variability along the routes represented by CTCs. The heights are proportional to the differences between the third and first quartiles of the flight levels and the widths to the mean distances of the aircraft positions to the CTCs.

(Italy), LJ (Slovenia), and LD (Croatia). Hence, by choosing this route, flight operators may reduce their expenses for navigation services.

Fig. 6 represents the route choices of the major flight operators. The columns with differently colored horizontal bar charts correspond to the flight operators labeled from FOP1 to FOP6. The rows correspond to the routes; the bar lengths are proportional to the number of times the routes were used by the flight operators. It can be seen that FOP4 always chose the cheapest route 2. Besides FOP4, route 2 was only sometimes flown by FOP1. Route 1 was preferred by all operators except FOP4, whereas route 5 was flown almost exclusively by FOP2.

To see the relative advantages and drawbacks of the routes, the analyst computes the statistics of various properties of the flights by the clusters using the data describing the actual flights. Surprisingly, route 5 crossing the north of Greece is longer than the others (except 8 and 9) in its main part, but the overall traveled paths are shorter, which can be attributed to more direct approaches to the landing runway. Hence, by using this route, flight operators can save fuel, also because it typically reaches higher flight levels than the others. Route 5 additionally has the second shortest flight duration, after route 8. However, route 5 is characterized by the highest arrival delays, although the departure

delays are similar to those for the other routes. Routes 1 and 2 have very similar statistics for path lengths, flight durations, and delays while the flight levels are higher on route 1, which can reduce fuel consumption.

The domain expert also wants to see how much the actual flights deviate from the central route lines and how the flight levels vary along the routes. To answer this question, the analyst derives statistics of the distances from the points of the actual flight trajectories to the corresponding points of the central trajectories of their clusters, as well as statistics of the flight levels. The results are shown on a map (Fig. 7). The flight level variability (namely, the differences between the third and first quartiles) is represented by the heights of the triangular symbols and the mean horizontal distances from the CTCs by the widths. To reduce the symbol overlapping, the visibility of the routes can be switched off and on. The domain expert observes that on route 2 the flights have higher altitude variations than on routes 1 and 5, and also there are large horizontal deviations in some places from the central line. The flights on route 1 were more coherent than on the other routes. The expert also notes the presence of horizontal deviations from the route central lines around the Strasbourg area, due to a military airspace that is frequently active and closed to civil aviation; this military area is also responsible for the remarkable deviation of route 9.

This study has revealed that, while there are flight operators striving to reduce the navigation costs, this is not the main route choice criterion for the majority of operators, who prefer the possibility to fly at higher altitudes as well as higher route stability (i.e., lower deviations), which lead to fuel economy. The objectives of the study were achieved by using filter-aware clustering of trajectories to reveal the major routes, tools for obtaining CTCs and summarizing characteristics of cluster members, and interactive visual displays supporting exploratory analysis of trajectories, clusters, and their attributes.

5.2 Exploring landing schemes of a major hub

The data in this case study consist of 5,045 trajectories (1,316,394 points) of the flights that landed at 5 different airports of London during 4 days from December 1 to December 4, 2016, and data describing the weather during this period. The analysis goals are: (1) extract the major approach routes into the airports of London, (2) investigate how the traffic that flows along these routes is separated in the 3D space, and (3) reveal the relationships between the use of the routes and wind parameters. The approach routes can be extracted by density-based clustering according to route similarity, which needs to be applied to the final parts of the trajectories; hence, the trajectory segments need to be filtered by the distances to the destinations. This is not sufficient, however. Many trajectories include holding loops (Fig. 8), which are made by aircraft as commanded by controllers while waiting for the possibility to land. The loops are not part of the proper landing approach and must be filtered out, otherwise they will strongly affect the clustering results: trajectories following the same route but differing in the number of loops will not be put in the same cluster.

To understand how to mark the loops in the data and filter them out, the analyst and the domain expert interactively explore the data and determine that one full loop takes approximately 5 minutes. So, the analyst applies a tool that computes for each trajectory position the sum of the turns in the next 5 minutes. For a position at the beginning of a loop, this sum should theoretically be about $\pm 360°$ (positive and negative values correspond to right and left turns, respectively). In practice, this value cannot be reached in discrete data, where not all turning points are present due to time gaps between the records. By interactive filtering and observing the results on a map display, the analyst and expert jointly ascertain that loop starts can be extracted using a threshold value of $\pm 240°$ for the summary turn over a 5-minute window. These positions of the loop starts and the following positions within the 5 minutes intervals are marked in the data as loops by creating a boolean attribute based on the current filter. To verify the result, the analyst builds the interactive visualization shown in Fig. 8, where the segments of the trajectories are colored according to the values of the boolean attribute just created; red color corresponds to the loops. To check the discrimination quality, the analyst switches on and off the red and blue parts of the trajectories and concludes that the loops have been identified correctly. Loops occurred in 1,484 trajectories (29.4%), including more than 50% of the flights that landed at Heathrow and about 5%-10% of the flights that landed at the other airports.

Fig. 8. The final parts of the trajectories of the flights that arrived to London. The holding loops are highlighted in red.

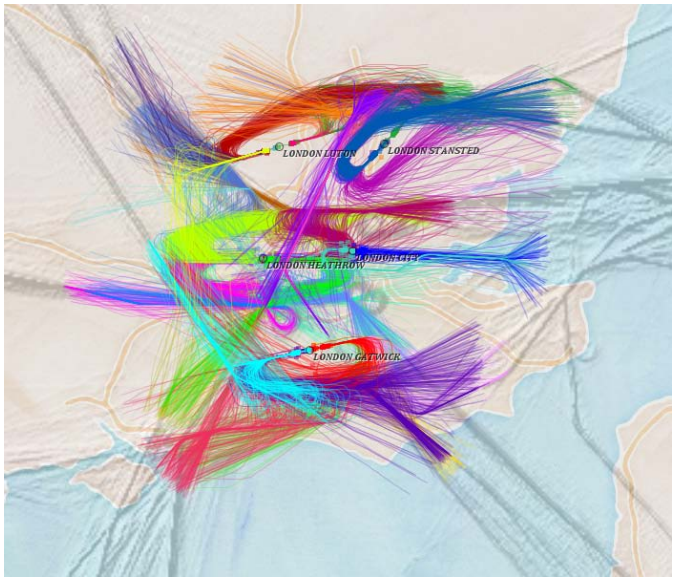

Fig. 9. 34 clusters representing the main approaches to the airports of London are represented by coloring of the relevant parts of the trajectories while the irrelevant parts are summarized in a density surface.

By filtering, the analyst hides the loops and selects the final parts of the trajectories starting from the 75 km distance to the destination. Then the analyst applies clustering by route similarity to the active parts. With $NR = 10km$ and $NN = 5$, the clustering separates very well different approach routes for all airports except Stansted. The analyst selects the subset of trajectories ending in Stansted and applies two steps of progressive clustering with $NR = 10km$ and $NR = 7.5km$ separately to this subset, which yields good, clean clusters of different approaches into Stansted. After merging the clustering results for all airports, there are in total 34 clusters (Fig. 9) including 4,628 trajectories (91.7%), while 417 trajectories (8.3%) are labeled as noise.

To see the temporal distribution of the arrivals via the routes revealed, the analyst creates a time histogram (Fig. 10, top) where the horizontal dimension represents the time span of the data divided into hourly intervals. For each interval, the total height of the corresponding bar shows the number of arrivals. The bars are divided into segments painted in the colors of the clusters; the segment heights are proportional to the numbers of arrivals via the respective routes. It is clearly seen that different routes were used on the first day than on the following

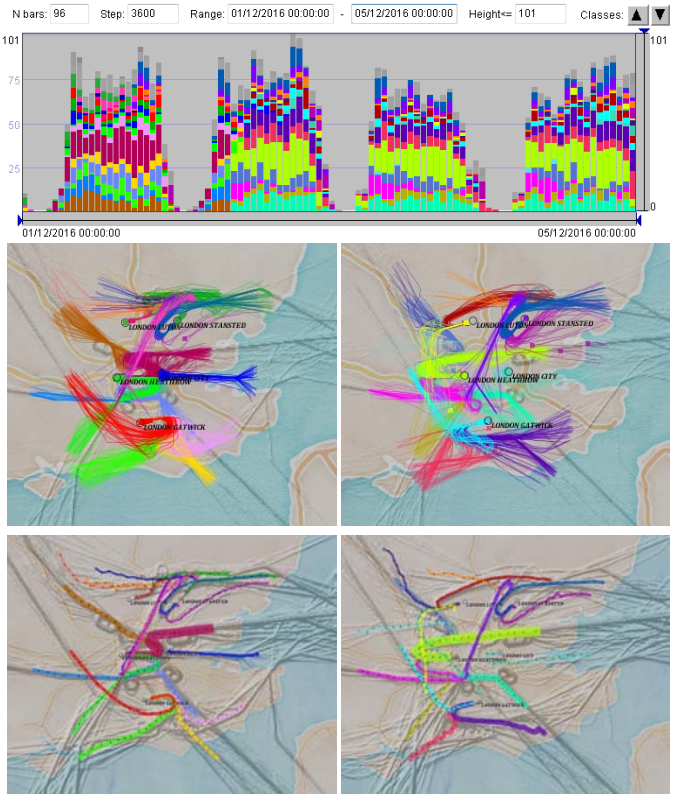

Fig. 10. Top: Bars in a time histogram show the counts of the flight arrivals in hourly intervals. Bar segments are painted in the colors of the route-based clusters the flights belong to. A difference between day 1 and days 2-4 is notable. Middle: The final parts of the flight trajectories in days 1 and 3 are colored according to the cluster membership. Bottom: The CTCs in days 1 and 3.

three days; a notable change happened at 10 AM on day 2. Using time filtering, the analyst can view the sets of approaches in different time intervals (Fig. 10, middle row, shows the approaches in the first and the third day). Time filtering allows the analyst to obtain sets of CTCs separately for different days. After canceling the time filter, the CTCs from different days can be seen simultaneously in two or more map windows. In Fig. 10, bottom, the CTCs from day 1 (left) and day 3 (right) are drawn on top of density maps of the overall traffic on these days, which show that the flights came to London from the same directions in both days but their final parts differed.

The routes used in the first day approached all airports except Stansted from the east while Stansted was approached from two opposite sides, northeast and southwest. Starting from about 10 AM on 02/12/2016, four airports were approached from the west and Stansted from the southwest, like is shown for day 3 in Fig. 10. Stansted differs from the other airports by the southwest-northeast orientation of the runway [5], while in the other airports it is west-east. The analyst explores in more detail the approaches into Stansted and finds that the routes approaching runway 22 from the northeast were used till about 18:25; after that, the approach routes from the southwest (runway 04) were used. This change is hard to see in the overall time histogram (Fig. 10, top), because the segments corresponding to Stansted are dominated by the segments corresponding to Heathrow. When only the flights to Stansted are interactively selected, the change is seen clearly.

The analyst compares the changes of the landing schemes to the changes in wind parameters, which consist of the magnitudes of two components of the wind vector, east-west and south-north. The wind was blowing from the northwest till the morning of day 2, then the west-to-east component changed to the opposite, and the wind magnitude in the east-to-west direction was increasing over days 2-4. The change of the sign of the east-west wind component explains the changes of the landing schemes and the approach routes in four airports out of five. For safety reasons, planes must take off and land into the wind [4]. When the wind blows from the west, as in day 1, the runway is approached from the east, and vice versa. The orientation of the runway in Stansted (44°/224° due North) makes it sensitive to the north-south component of the wind, which is less important for the other airports. In the afternoon of day 1, the magnitude of the north-to-south component substantially exceeded the magnitude of the west-to-east component. This caused the change of the landing scheme so that the runway was approached from the southwest to face the northern wind.

To investigate the separation of the air traffic flows, the analyst and the domain expert use the maps of the approach routes (Fig. 10, bottom) in combination with 3D displays of the routes (Fig. 1) based on NASA World Wind [6]. All routes in their final parts align with the orientation of the runways of the destination airports. The maps show that routes to the same airport from different directions join together before landing, and the 3D view shows that the routes also join in the vertical dimension. The routes to different airports that cross or partly go together in the 2D view are separated vertically. Thus, in day 1, three routes going from the southwest northwards into Luton and Stansted appear as crossing some routes into Heathrow on the 2D map, but the 3D view shows that they go at much higher altitudes than the routes to Heathrow. The route into Luton, which appears as going partly together with two routes to Stansted on the 2D map, goes above these routes in the 3D view. Generally, the 3D view shows that vertical separation occurs in all cases when routes into different airports are not separated in 2D.

By request of the domain expert, the analyst explores how the holding loops are distributed among the approach routes. The analyst changes the filter of the trajectory segments to make the loops active and the remaining parts inactive and aggregates the data from the loops by the routes. The largest absolute number of flights with loops occurred on the most actively used route to Heathrow in which the flights coming from the east approached the airport from the west (colored in greenish yellow in Fig. 10). The highest proportions of flights containing holding loops were on the routes coming to Heathrow from the northwest, west, and southwest and landing from the east in day 1 and from the west in the following days. The mean and median durations of the looping movement were from 5 to 10 minutes, which means 1 or 2 loops per flight, while the maximal durations on some routes (all into Heathrow) reached 20-24 minutes (4-5 loops).

In this case study, the RACAT workflow was used to reveal the main approach routes to different airports irrespective of the presence of holding loops. It was analyzed how the use of the routes changed over time in relation to the wind parameters. The extraction of the routes also facilitated the exploration of the traffic flow separation schemes. The domain expert acknowledged these capabilities as very useful and novel for operational analysis and modeling for increased predictability.

5.3 Reconstructing a generalized air traffic network

A configuration is a particular division of an airspace region into sectors, such that each sector is managed by two air traffic controllers. For a given time interval, a region R is divided into N sectors depending, on the one hand, on the expected demand (i.e., the number of the flights that will be carried out within and across R), and, on the other hand, on the available number of controllers. The expected demand is estimated based on the flight plans that are sent by the flight operators (airlines) to traffic flow managers before the beginning of an operation day. Flights are mostly done along fixed navigation routes, named airways. There may be multiple ways to divide a region R into N sectors. The choice among them depends on the expected traffic intensities (a.k.a. traffic flows) on different routes.

The current practices of choosing configurations by flow managers are not transparent as they involve human decision makers with their tacit knowledge and preferences. Air traffic researchers are interested in building a model that could explain and predict the configuration choices. For this purpose, the configuration that was applied in a region in each time interval needs to be matched with the expected traffic flows on different routes and in different directions. A suitable approach is to extract the major routes existing in the region and compute the number and temporal density (frequency) of the expected flights on each route by time intervals. The major routes can be extracted from a set of flight

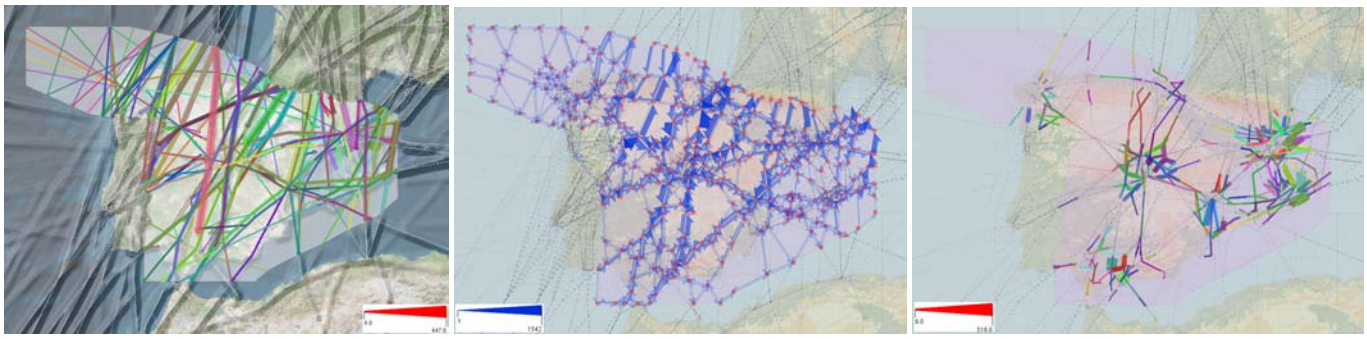

Fig. 11. Left: The major routes in the upper air subspace of Spain are represented by the central trajectories of flight clusters. The colors differentiate the clusters and the line widths are proportional to the cluster sizes. Middle: Planned flights have been aggregated by a generalized air traffic network constructed on the basis of the extracted major routes. The widths of the lines are proportional to the magnitudes of the traffic flows along the links of the network. Right: The major routes in the lower air subspace of Spain.

plan-based trajectories by applying clustering to their relevant parts.

Apart from fitting in or crossing the region under analysis, the relevant parts of the trajectories required for the route extraction are selected based on the flight levels of the trajectory points. The airspace of each country in Europe is divided in the vertical dimension into lower and upper subspaces with the flight levels below and above 245. The upper subspace corresponds to the en-route parts of the flights and the lower subspace to the approach and take-off areas surrounding airports. As the subspaces are controlled by different rosters, the division into sectors is done separately for the regions in the upper and lower subspaces. The sets of possible configurations are different for the upper and lower regions because the respective routes differ. Hence, it is necessary to extract the major routes separately from the trajectory parts lying in the upper and lower subspaces. This requires application of the clustering procedure to different relevant parts of the trajectories.

This case study focuses on the airspace of Spain. The analyst applies the progressive clustering procedure first to the parts of flight trajectories with flight levels over 245 and then to the parts with flight levels below 245. In Fig. 11, left and right, the clusters obtained for the upper and lower subspaces are represented by CTCs reflecting the major routes. Line widths are proportional to the cardinalities of the clusters. The domain expert sees that the detected clusters correspond quite well to the actual main operational flows, but the picture of the upper space routes slightly differs from what was expected. Usually, the dominant flows are between France and Madrid. While they are still very prominent, the attention is attracted to higher than usual flows between the UK and the south of Spain as well as the flows from France and Germany to Balearic Islands. The expert easily finds an explanation: the time period under study includes the Easter holidays, when the south of Spain and Balearic Islands were attractive tourist destinations.

There are two possible ways in which the extracted major routes can be used for the aggregation of traffic data and estimation of the demands by time intervals. The first way is to assign each flight to the best matching route (this can be done using the "route similarity" function) and count the flight number per route and time interval. The second way is to construct a generalized traffic network by treating the intersection points of the routes as nodes and the route segments connecting them as links. The traffic data can then be aggregated by the links of such a network, as illustrated in Fig. 11-middle. The red dots on the map mark the network nodes, and the half-arrow symbols between them show the aggregated traffic flows, the line widths being proportional to the flow magnitudes. The domain expert finds that the map shown in Fig. 11-middle meaningfully summarizes the overall traffic and reflects very well the known patterns of flows.

The major routes in the lower subspace do not make a network, as can be seen from Fig. 11-right. The domain expert explains that the lower air subspace consists of mostly disjoint terminal maneuvering areas (TMAs) that connect airports to the upper subspace. Hence, to calculate the demands for the lower air subspace, the flights can be aggregated by the major routes but not by a network.

In this case study, selection and clustering of relevant parts of flight trajectories allowed the analyst to extract the major flight routes in the upper and lower subspaces of Spain. These routes provide a suitable basis for the calculation of the demands for the air navigation services.

6 Discussion and Conclusion

Our case studies have convinced the domain expert that the proposed techniques are effective for the chosen classes of tasks. In the expert's opinion, the performed analyses are highly innovative in the ATM domain and deserve being developed into full-fledged general procedures for solving the classes of problems represented by the case studies. The expert also expressed his belief that the techniques have a great potential for application to other classes of problems in the domain.

Generally, the case studies demonstrate that the RACAT workflow can be flexibly used for a variety of analysis tasks in which different parts of trajectories are relevant. We would like to point out that the approach is not specific to aviation but applicable to other domains where trajectories of moving objects need to be analyzed. Thus, in the ground transportation domain [12], analysis tasks may require clustering of trajectories according to their parts on highways, or during rush hours, or within congested areas. In football analysis [11], relevant parts of players' trajectories may need to be clustered taking into account ball possession, position on the pitch, and direction and speed of the movement. As the notion of relevance may change throughout the analysis process, interactive filtering tools should enable dynamic modification of the relevance masks, and the clustering needs to be done without destroying the original trajectories due to extracting only relevant parts.

The workflow can be implemented based on any clustering algorithm that permits the use of specific distance functions for trajectories, and any distance function that works at the level of trajectory points or segments. The distance function needs to be modified so that only active points or segments are taken into account. This approach is applicable to any set of trajectories. However, the idea of representing clusters by central trajectories is less general: it makes sense only for clusters of trajectories having similar geometric shapes, i.e., following similar routes. When trajectories are clustered based on other criteria, there may be no common route that can be represented by a single line. In such cases, clusters of trajectories can be summarized in other ways, e.g., by using space tessellation and transforming trajectories into moves between cells [10].

In this paper, we have proposed a set of general techniques and visualization guidelines applicable to a wide range of tasks in analysis of movement data. The techniques were tested in case studies with complex real-world data and non-trivial analysis tasks of high practical importance and proved their effectiveness. They can be recommended for use in various domains requiring analysis of movement data.

Acknowledgments

The authors wish to thank S.Rinzivillo (CNR, Pisa, IT), C.Navarra and K.Vrotsou (Univ. Lincöping, SE), D.Knodt (Fraunhofer IAIS, DE), D.Scarlatti (Boeing Research, Madrid ES), R.Herranz and R.Marcos (Nommon, Madrid ES). This work was supported in part by EU in project datAcron (grant 687591) and by SESAR JU in projects DART (grant 699299) and INTUIT (grant 699303).

REFERENCES

[1] 30,000 flights covering 25 million miles: Beautiful video reveals an entire day of european air travel in just two minutes. http://www.dailymail.co.uk/sciencetech/article-2579190. Accessed: 24.03.2017.

[2] FlightAware. http://www.flightaware.com. Accessed: 24.03.2017.

[3] FlightRadar24. https://www.flightradar24.com. Accessed: 24.03.2017.

[4] Heathrow wind direction. http://www.heathrow.com/noise/heathrow-operations/wind-direction. Accessed: 24.03.2017.

[5] London stansted airport. https://skyvector.com/airport/EGSS/London-Stansted-Airport. Accessed: 24.03.2017.

[6] NASA World Wind. https://worldwind.arc.nasa.gov/. Accessed: 24.03.2017.

[7] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011.

[8] G. H. Albrecht, H. T. Lee, and A. Pang. Visual analysis of air traffic data using aircraft density and conflict probability. In *Infotech@ Aerospace 2012*. 2012.

[9] H. Alt and M. Godau. Computing the Frchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 05(01n02):75–91, 1995. doi: 10.1142/S0218195995000064

[10] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel. *Visual Analytics of Movement*. Springer, 2013. doi: 10.1007/978-3-642-37583-5

[11] G. Andrienko, N. Andrienko, G. Budziak, J. Dykes, G. Fuchs, T. von Landesberger, and H. Weber. Visual analysis of pressure in football. *Data Mining and Knowledge Discovery*, 2017. doi: 10.1007/s10618-017-0513-2

[12] G. Andrienko, N. Andrienko, W. Chen, R. Maciejewski, and Y. Zhao. Visual analytics of mobility and transportation: State of the art and further research directions. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–18, 2017. doi: 10.1109/TITS.2017.2683539

[13] G. Andrienko, N. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel. Scalable analysis of movement data for extracting and exploring significant places. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1078–1094, July 2013. doi: 10.1109/TVCG.2012.311

[14] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 3–10, Oct 2009. doi: 10.1109/VAST.2009.5332584

[15] N. Andrienko, G. Andrienko, L. Barrett, M. Dostie, and P. Henzi. Space transformation for understanding group movement. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2169–2178, Dec 2013. doi: 10.1109/TVCG.2013.193

[16] N. Andrienko, G. Andrienko, E. Camossi, C. Claramunt, J. M. Cordero-Garcia, G. Fuchs, M. Hadzagic, A.-L. Jousselme, C. Ray, D. Scarlatti, and G. Vouros. Visual exploration of movement and event data with interactive time masks. *Visual Informatics*, 1(1):25 – 39, 2017. doi: 10.1016/j.visinf.2017.01.004

[17] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, June 1999. doi: 10.1145/304181.304187

[18] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, vol. 10, pp. 359–370. Seattle, WA, 1994.

[19] J. Buchmüller, H. Janetzko, G. Andrienko, N. Andrienko, G. Fuchs, and D. A. Keim. Visual analytics for exploring local impact of air traffic. *Computer Graphics Forum*, 34(3):181–190, 2015. doi: 10.1111/cgf.12630

[20] S. Buschmann, M. Trapp, and J. Döllner. Animated visualization of spatial–temporal trajectory data for air-traffic analysis. *The Visual Computer*, 32(3):371–381, 2016. doi: 10.1007/s00371-015-1185-9

[21] L. Delgado. European route choice determinants. examining fuel and route charge trade-offs. In *Eleventh USA/Europe Air Traffic Management Research and Development Seminar (ATM2015)*, 2015.

[22] M. Enriquez and C. Kurcz. A simple and robust flow detection algorithm based on spectral clustering. In *ICRAT Conference*, 2012.

[23] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareiro, and A. Telea. Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2364–2373, Dec 2011. doi: 10.1109/TVCG.2011.233

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press, 1996.

[25] N. Ferreira, J. T. Klosowski, C. E. Scheidegger, and C. T. Silva. Vector field k-means: Clustering trajectories by fitting multiple vector fields. *Computer Graphics Forum*, 32(3pt2):201–210, 2013. doi: 10.1111/cgf.12107

[26] M. Gariel, A. N. Srivastava, and E. Feron. Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1511–1524, Dec 2011. doi: 10.1109/TITS.2011.2160628

[27] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal*, 20(5):695–719, Oct. 2011. doi: 10.1007/s00778-011-0244-8

[28] M. Harrower, A. L. Griffin, and A. MacEachren. Temporal focusing and temporal brushing: assessing their impact in geographic visualization. In *Proceedings of 19th International Cartographic Conference, Ottawa, Canada*, vol. 1, pp. 729–738, 1999.

[29] M. Harrower, A. MacEachren, and A. L. Griffin. Developing a geographic visualization tool to support earth science learning. *Cartography and Geographic Information Science*, 27(4):279–293, 2000.

[30] B. K. P. Horn. Hill shading and the reflectance map. *Proceedings of the IEEE*, 69(1):14–47, Jan 1981. doi: 10.1109/PROC.1981.11918

[31] C. Hurter, R. Alligier, D. Gianazza, S. Puechmorel, G. Andrienko, and N. Andrienko. Wind parameters extraction from aircraft trajectories. *Computers, Environment and Urban Systems*, 47:28 – 43, 2014. doi: 10.1016/j.compenvurbsys.2014.01.005

[32] C. Hurter, Y. Brenier, J. Ducas, and E. L. Guilcher. Cap: Collaborative advanced planning, trade-off between airspace management and optimized flight performance: Demonstration of en-route reduced airspace congestion through collaborative flight planning. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1–9, Sept 2016. doi: 10.1109/DASC.2016.7777947

[33] C. Hurter, M. Serrurier, R. Alonso, G. Tabart, and J.-L. Vinot. An automatic generation of schematic maps to display flight routes for air traffic controllers: Structure and color optimization. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pp. 233–240. ACM, New York, NY, USA, 2010. doi: 10.1145/1842993.1843034

[34] C. Hurter, B. Tissoires, and S. Conversy. Fromdady: Spreading aircraft trajectories across views to support iterative queries. 15(6):1017–1024, 2009. doi: 10.1109/TVCG.2009.145

[35] S. Jang, N. Elmqvist, and K. Ramani. Motionflow: Visual abstraction and aggregation of sequential patterns in human motion tracking data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):21–30, Jan 2016. doi: 10.1109/TVCG.2015.2468292

[36] H. Jeung, M. L. Yiu, and C. S. Jensen. *Trajectory Pattern Mining*, pp. 143–177. Springer New York, New York, NY, 2011. doi: 10.1007/978-1-4614-1629-6_5

[37] T. Klein, M. van der Zwan, and A. Telea. Dynamic multiscale visualization of flight data. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 1, pp. 104–114, Jan 2014.

[38] A. Koblin. Flight patterns. http://www.aaronkoblin.com/work/flightpatterns/. Accessed: 24.03.2017.

[39] O. D. Lampe and H. Hauser. Interactive visualization of streaming data with kernel density estimation. In *2011 IEEE Pacific Visualization Symposium*, pp. 171–178, March 2011. doi: 10.1109/PACIFICVIS.2011.5742387

[40] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pp. 593–604. ACM, New York, NY, USA, 2007. doi: 10.1145/1247480.1247546

[41] M. Nanni and D. Pedreschi. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289, 2006. doi: 10.1007/s10844-006-9953-7

[42] N. Pelekis, G. Andrienko, N. Andrienko, I. Kopanakis, G. Marketos, and Y. Theodoridis. Visually exploring movement data via similarity-based analysis. *Journal of Intelligent Information Systems*, 38(2):343–391, 2012. doi: 10.1007/s10844-011-0159-2

[43] A. Perer and F. Wang. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pp. 153–162. ACM, New York, NY, USA, 2014. doi: 10.1145/2557500.2557508

[44] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3-4):225–239, 2008. doi: 10.

1057/PALGRAVE.IVS.9500183

[45] D. Sacha, F. Al-Masoudi, M. Stein, T. Schreck, D. A. Keim, G. Andrienko, and H. Janetzko. Dynamic visual abstraction of soccer movement. *Computer Graphics Forum*, 36(3):(accepted), 2017.

[46] M. Sakr, G. Andrienko, T. Behr, N. Andrienko, R. H. Güting, and C. Hurter. Exploring spatiotemporal patterns by integrating visual analytics with a moving objects database system. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pp. 505–508. ACM, New York, NY, USA, 2011. doi: 10.1145/2093973.2094060

[47] E. Salaun, M. Gariel, A. E. Vela, and E. Feron. Aircraft proximity maps based on data-driven flow modeling. *Journal of Guidance, Control, and Dynamics*, 35(2):563–577, 2012.

[48] T. Schreck, J. Bernard, T. von Landesberger, and J. Kohlhammer. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization*, 8(1):14–29, 2009. doi: 10.1057/ivs.2008.29

[49] M. J. van Kreveld, M. Löffler, and F. Staals. Central trajectories. *CoRR*, abs/1501.01822, 2015.

[50] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings 18th International Conference on Data Engineering*, pp. 673–684, 2002. doi: 10.1109/ICDE.2002.994784

[51] K. Vrotsou, H. Janetzko, C. Navarra, G. Fuchs, D. Spretke, F. Mansmann, N. Andrienko, and G. Andrienko. Simplifly: A methodology for simplification and thematic enhancement of trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):107–121, Jan 2015. doi: 10.1109/TVCG.2014.2337333

[52] C. Ware, R. Arsenault, M. Plumlee, and D. Wiley. Visualizing the underwater behavior of humpback whales. *IEEE Computer Graphics and Applications*, 26(4):14–18, July 2006. doi: 10.1109/MCG.2006.93

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