A Trajectory Clustering Framework to Analyse Air Traffic Flows

Luis Basora, Jérôme Morio, Corentin Mailhot

A Trajectory Clustering Framework to Analyse Air Traffic Flows

Luis Basora and Jérôme Morio
Information Processing and Systems Department
ONERA, The French Aerospace Lab - 31055
Toulouse, France
Email: luis.basora@onera.fr, jerome.morio@onera.fr

Corentin Mailhot
ENAC
DGAC French Civil Aviation
Toulouse, France
Email: corentin.mailhot@aviation-civile.gouv.fr

***Abstract*—This paper describes a framework to automatically identify air traffic flows from a set of trajectories by using a clustering algorithm. The framework offers two methods to cluster trajectories, each one using a different distance/similarity measure between trajectories. Results and performance characteristics of both methods are compared by applying them to real trajectories over a French Area Control Center. The framework can output statistics and figures for flow analysis and its use is facilitated by the relatively low number of parameters to be provided by the user. Its aim is to help support the SESAR vision of flow-centric operations by being integrated into Air Traffic Management tools, e.g. for airspace design/management or for analysis of traffic patterns in a free route environment.**

I. Introduction

In Air Traffic Management (ATM), massive amounts of data are available to feed data-driven models for real-time decision support or to identify behaviours or patterns relevant to the operational performance of the system in post-event analysis. In particular, sources such as radars, Automatic Dependent Surveillance-Broadcast (ADS-B) or trajectory prediction models generate samples of data points representing trajectories that can then be clustered to identify traffic flows. Trajectory clustering algorithms could be useful in the context of the flow-centric operations vision of SESAR, as described in the European ATM Master Plan [1], by being integrated into tools supporting airspace design/management, complexity management, etc. This is particularly true in a free route environment, where the capacity to understand traffic flows is even more necessary as fixed routes will no longer structure the traffic.

Clustering is a widely used data analysis technique in the statistics/machine learning field. It is about grouping entities with similar characteristics together, so the notion of similarity/distance is essential to the problem. A number of clustering algorithms have been reported in the literature, such as k-means [2], BIRCH [3], DBSCAN [4] and OPTICS [5], all of which are oriented towards the clustering of point data. Even though trajectories have a functional nature (curves), these algorithms can still be applied since trajectory data is available in the form of samples of data points.

For instance, in a recent paper [6], DBSCAN shows promising results when applied to the characterisation of traffic flows based on recorded radar tracks in the terminal airspace of the New York Metro region. This algorithm has the capability to handle noise/outlier data and does not require the number of clusters to be provided as an input parameter. Previously, in [7], another framework is described based on DBSCAN and k-means to analyse the patterns of traffic over the Northern California terminal area. In both of these studies, however, no distance/similarity measures between trajectories is provided, i.e. the clustering is based solely on the density of the individual trajectory points.

Alternatively, in [8] and [9], two different methods are presented to cluster trajectory segments rather than trajectory points. The former (TRACLUS) implements a variant of DBSCAN to cluster the segments but has never been successfully applied to air traffic as far as we know. The latter is designed to identify air traffic flows in the National Airspace System (NAS), but it is limited to 2D and based on the development of specific algorithms for incremental clustering requiring a non-obvious parameter setting by the user.

Another interesting study that examines the problem of clustering air traffic trajectories is reported in [10] where a spectral clustering approach is used to consider the temporal characteristics of the air traffic flows in the US. However, the procedure to take into account the shape of the trajectories is not evident and the curvature of the clustering centroids might not always be very realistic from an operational point of view. Other approaches such as in [11] relies on a graph structure like a road network, which is not adapted to ATM because of the direct routes often allocated by ATC for tactical reasons or when applied to a free route environment.

More recently, in order to overcome the limitations of some of these techniques, an approach based on entropy minimisation and Lie group modelling has been proposed in [12]. Unfortunately, only partial results are available at the moment of writing this paper and the implementation of the algorithms is fairly complex.

In this paper, we propose a new framework for air traffic flow analysis based on an improved version of DBSCAN called Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [13] which is able to manage clusters of different densities with a single input parameter. Two methods based on two different distance functions between trajectories, Euclidean Distance (ED) and Symmetrized

Segment-Path Distance (SSPD) [14] can be selected by the user. These distances offer a trade-off between accuracy and runtime. In addition to the clustering capability, the framework provides several components to filter and interpolate trajectories, compute basic flow statistics and export clusters in Keyhole Markup Language (KML). The framework can be useful to analyse traffic patterns in a wide range of operational scenarios both in en-route and approach. In addition, its implementation and integration into ATM applications is facilitated by the public availability in several languages of some of the algorithms like HDBSCAN [15].

This paper is organised as follows. Section II details the methodology and the different components of the framework. Section III presents the results and the comparison of the two methods by using DDR2 [16] trajectory data provided by Eurocontrol over the Reims Area Control Center (ACC). Finally, section IV summarizes and identifies some ideas for future work and potential applications.

II. The Framework

The framework is designed to satisfy the following requirements:

1) It shall be possible to filter trajectories geographically, by airspace, by altitude and in time.
2) It shall be possible to cluster in 2 or 3 dimensions, optionally taking the heading into account as well.
3) The clustering algorithm shall allow for variable trade-off between computing time and quality of the resulting clusters so as to be useful in scenarios with either large or small sets of trajectories.
4) The clustering algorithm shall work within an area where traffic density distribution is not uniform.
5) The clustering algorithm shall work with noise data and be able to identify outliers.
6) The clustering algorithm should require a reduced number of input parameters with clear ATM operational significance.
7) For each cluster it shall be possible to compute the centroid in addition to the set of associated trajectories.
8) For each cluster it shall be possible to compute statistics such as the flow rate, flight distribution per origin/destination, average distance and heading of the cluster trajectories.

Figure 1 shows the framework architecture with the two clustering methods in the center and the pre-processing and post-processing steps. The user can choose the method and the associated parameters. The clustering method based on ED works faster than SSPD at the possible expense of a lower quality clustering result. The specific steps of the framework depend on the selected distance and are described in the following sub-sections.

The framework is implemented in Python 3 with the following Python libraries: scikit-learn 0.18.2, RDP 0.8, Hdbscan 0.8.8, Pyproj 1.9.5.1, Planar 0.4. The HDBSCAN library is highly optimised, but otherwise no parallelism, multi-threading or code optimisation has been used so far in areas such the SSPD implementation where considerable gains of performance can be expected.

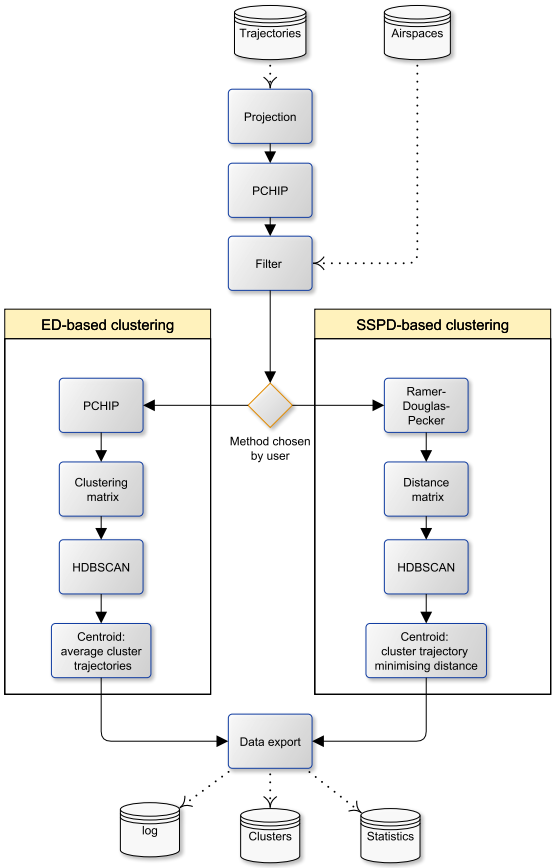


Figure 1. Trajectory Clustering Framework process for both the ED and SSPD-based distance methods. The framework automatically identifies the flows from a set of trajectories by applying the HDBSCAN clustering algorithm and generates statistics for flow analysis.

A. Preprocessing

Trajectory datasets contain 3D position samples (latitude, longitude, altitude) for each trajectory and possibly other information like speed and heading. Latitude/longitude coordinates are projected in order to facilitate distance computation. Speed is not used by the clustering algorithms and heading is optional in the ED-based clustering method to ensure flows in opposite directions are separated into different clusters. If heading information h is required but not part of the dataset, it is calculated for each trajectory segment from geographic coordinates $[(lat1, lon1), (lat2, lon2)]$ and $\Delta lon = lon2 - lon1$ by:

$$h = (\arctan(a) + 360) \ \% \ 360 \tag{1}$$

where

$$a = \frac{\cos(lat2)\sin(\Delta lon)}{\cos(lat1)\sin(lat2) - \sin(lat1)\cos(lat2)\cos(\Delta lon)}$$

It is important to note that position samples can be irregularly distributed along the trajectory as shown in Figure 2.

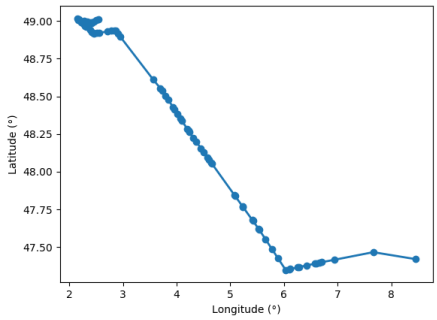


Figure 2. Trajectory discretisation example

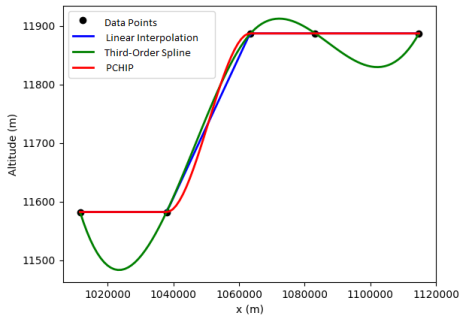


Figure 3. Comparison of interpolation methods

Many data points can be removed as they are redundant, whereas other positions need to be interpolated.

We use the Ramer-Douglas-Peucker (RDP) [17] [18] algorithm to remove redundant trajectory information, but only in the case of SSPD where considerable runtime gains can be expected. Further details on the application of this algorithm are given in section II-B2.

Regarding the interpolation method, there are several methods available from simple linear interpolation to more sophisticated polynomial or spline interpolation. In the framework, the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) [19] [20] method is used as it better takes into account the operational reality of air traffic trajectories. As shown in the example of the vertical profile of a flight in Figure 3, linear interpolation is too abrupt, whereas the order 3 spline ensures the continuity but with oscillations (overshoot). With PCHIP, however, we get a smoother curve much more representative of a true flight trajectory.

Finally, as for the filtering capability, in addition to a an altitude range or a time window, the framework allows the user to chose either a bounding box around the geographic area to be studied or a set of airspaces (e.g. ACC or sector). In the latter case, the convex hull is computed for the set of airspaces.

B. Clustering

Once the preprocessing phase is complete and trajectories have been properly prepared, we can apply the HDBSCAN algorithm to them to identify the traffic flows. Compared to DBSCAN, HDBSCAN presents several improvements. First, it only requires the user to define the minimum cluster size, i.e. the minimum number of points (trajectories in our case) to form a cluster. Secondly, HDBSCAN works better than DBSCAN for data with varying density, which is usually the case for air traffic trajectories. Thirdly, HDBSCAN prevents the ”bridge effect” between two clusters because of a single or a few data points in the middle of the two clusters (potentially gluing them together) by considering these points as noise.

The choice of an appropriate distance/similarity measure between trajectories is as important as the choice of the clustering algorithm. There are quite a few functions to compute trajectory distances reported in the literature, each one offering a different trade-off between computation time, accuracy and sensitivity to noise data, from the simple ED to the more sophisticated and complex ones like Fréchet and Hausdorff. As our framework is to be used in a wide range of use case scenarios, it cannot be based on a single metric. Also, we need a similarity measure that accounts for both the physical distance and the global shape between trajectories, i.e. time or other trajectory characteristics will not be considered.

We decided to choose the ED function since it is computationally fast and so potentially useful for scenarios involving a large number of trajectories, such as the ones covering wide geographic areas or large time windows. For scenarios where accuracy is more important and execution time less of an issue, SSPD distance should be used instead.

HDBSCAN labels each trajectory with a cluster number (outliers are labelled with a -1) and measures the strength of the cluster membership for each trajectory in the cluster with a probability. In this framework, a cluster is defined by three attributes: a cluster number, the list of trajectories belonging to this cluster (with probability equal to 1), and a representative trajectory or centroid (this last attribute is computed after the execution of HDBSCAN).

In the following subsections, the specificities of the application of HDBSCAN with each distance are detailed.

1) ED-based Clustering: The first step in this method is to build a matrix where each row represents a whole trajectory built from the sequence of trajectory points. This requires all the trajectories to have exactly the same number of points, which is indeed the case as we interpolate the trajectories according to the number of points chosen by the user. This is done at least once in the pre-processing phase before filtering, and again if the trajectories are clipped as a result of a filter being applied.

For instance, in the case of a 3D clustering with heading, N trajectories and n points per trajectory, the matrix would be defined as:

$$\begin{bmatrix} x_{11}y_{11}z_{11}h_{11}x_{12} & \dots & x_{1n}y_{1n}z_{1n}h_{1n}x_{1n} \\ \vdots & \ddots & \vdots \\ x_{N1}y_{N1}z_{N1}h_{N1}x_{N2} & \dots & x_{Nn}y_{Nn}z_{Nn}h_{Nn}x_{Nn} \end{bmatrix} \quad (2)$$

where $x_{ij}y_{ij}$ is the projected position, z_{ij} the altitude (in meters) and h_{ij} the heading computed as in (1). In order for

these elements to have the same weight, we standardise each value in the matrix by subtracting the mean and dividing by the variance.

Once these steps are concluded, we can apply the HDBSCAN algorithm with the following two parameters: the minimum number of trajectories per cluster (user parameter) and the clustering matrix built as in (2). ED is computed from the clustering matrix by the HDBSCAN itself as it is part of the standard distances already implemented in the algorithm.

The last step is to calculate the centroid of each cluster as the mean of the trajectories of the cluster.

2) SSPD-based Clustering: From a mathematical point of view, SSPD should be considered as a similarity measure rather than a distance or metric, because although it is symmetric, the triangle inequality property is not satisfied. Compared to the Fréchet distance, SSPD does not consider the speed of the trajectory since it is purely geometrical. On the other hand, if two trajectories are similar during the en-route phase because they share the same air routes and diverge only in the final phase of the flight at the terminal area, distance would be over-estimated by both Hausdorff and Fréchet. This is not the case with SSPD, because it better takes into account the global difference in the shape of both trajectories. Also, it presents a good trade-off between the simpler and faster ED and the more complex and time-consuming Hausdorff and Fréchet distances. However, with the SSPD distance it may be difficult to separate trajectories that are geographically close, similar in shape and length but having opposite directions. This may be less of an issue when the clustering is performed in 3D, as air routes for instance are designed to vertically separate flows in opposite directions.

After the pre-processing phase where trajectories are projected, interpolated and finally filtered for the area of interest, there is no need to interpolate the filtered trajectories as opposed to in the ED case. This is because SSPD does not require the trajectories to have the same number of points. However, the SSPD distance being much slower to compute than the ED, the RDP algorithm is applied first to remove redundant trajectory points.

The RDP algorithm can be particularly effective when applied to the en-route phase of the trajectory where aircraft follow long great circle segments linking the flight plan waypoints. Nevertheless, even during the cruise phase, aircraft do turn to follow jet routes and take advantage of favourable winds, avoid hotspot areas, etc. Therefore, we need to use this algorithm carefully enough so as not to exclude too many (if any) turning points. RDP accepts a threshold parameter allowing for the simplification, to a greater or lesser degree, of the trajectory, which can be specified by the user as a framework parameter. With a threshold of 200 meters we notice already an important reduction of redundant points for some trajectories like in Figure 4.

Once RDP has been applied to the trajectories and the distance matrix is computed, we can run HDBSCAN to perform the clustering. Afterwards, we finally obtain the cluster centroids by choosing the trajectory in the cluster that minimises the distance with the other trajectories in the cluster. Therefore, with this method, centroids are true trajectories and not virtual trajectories like with ED.

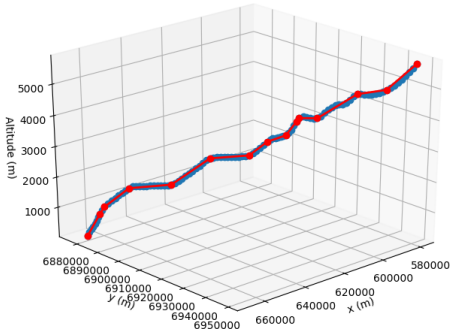


Figure 4. Application of Ramer-Douglas-Peucker (RDP) algorithm example: points reduced from 100 to 15.

C. Post-processing

The objective of this phase is to generate from the clustering results all the necessary outputs to enable the user to analyse the structure of the traffic represented by the clusters. In particular, the framework can generate:

1) 3D and 2D plots of the centroids indicating visually the flow intensity and direction.
2) Histograms to show the distribution of trajectories per cluster.
3) A set of statistics per cluster:
   - average length/altitude/heading,
   - flow rate,
   - flight distribution per origin/destination pair, origin and destination.
4) A KML file to display in detail the clusters as well as other related ATM structures such as the airspaces and the air routes.

III. Results

We assess the performance of both clustering methods by applying them to one day of traffic (26 June 2015) over the Reims ACC area. The dataset provided by Eurocontrol in DDR2 format contains 9442 trajectories. A bounding box with coordinates $[(46.7°, 1.328889°), (51.116667°, 8.218611°)]$ is defined around the Reims area to filter the trajectories geographically and different evaluations are performed on several altitude intervals. The coordinates are projected with Lambert-93 which is the official projection in Metropolitan France.

The objective is to identify the flows over Reims ACC and for that we need to specify the minimum number of trajectories (minimum cluster size) for a flow to be considered as such. This is the main clustering parameter to be defined by the user, and its value is highly application-dependent. Our purpose being purely the assessment of the algorithms, we set it up so that major flows can be identified.

The computer used in the evaluations is an Intel®Xeon Quad-Core 2.80GHz processor with 6GB of memory.

TABLE I
EXPERIMENTAL SETTINGS

	ED	SSPD	ED vs SSPD
Dimension	3	3	2
Min. Alt (ft)	None	31500	30000
Max. Alt (ft)	None	34500	50000
Interpolation Points	100	100	100
Min. Length (NM)	20	20	20
Min. Cluster Size	50	10	50

A. *ED-based clustering results*

This first experiment covers the complete volume of traffic in Reims ACC and is performed with the parameters specified in the first column of Table I.

We want to identify flows of at least 50 trajectories. Even though increasing the number of points after interpolation may improve the clustering accuracy, we have found 100 points to be a reasonable value. We also filter out any trajectory shorter than 20 NM, which reduces the number of trajectories to 9307.

PCHIP interpolation has the longest computation time, taking 99 seconds against 36 seconds for HDBSCAN. In total, outputs are generated in about 3 minutes, which is acceptable for a scenario with more than 9000 trajectories.

Figure 5 shows the 2D and 3D views of the 38 cluster centroids, where the thickness of the centroid lines is proportional to the intensity of the flow as indicated in the flight distribution per cluster figure. The algorithm has identified both en-route and terminal flows, the latter ones mainly around Paris Charles-de-Gaulle or Paris-Orly (Paris coordinates are $49°, 2.3°$) terminal area. We can use the KML output in Figure 6 to better visualise the centroids for the flows departing from and approaching Paris.

The number of outliers (32%) is high, but it depends on the size of the cluster (in our case 50) as shown in Figure 7. Also, it can be explained by the fact that the chosen day was the one with the largest volume of traffic for 2015 in Reims. This particularly high traffic density may have induced a higher complexity and required exceptional measures to resolve conflicts or hotspots.

B. *SSPD-based clustering results*

In this case, we use the SSPD-based clustering method to analyse the flows in the LFEEXR sector, with the parameters in the second column of Table I.

After filtering out the flights not entering this sector, a total number of 349 trajectories remain. The RDP algorithm, with a threshold of 200 meters, is next applied to remove redundant trajectory points. Then the SSPD distance is calculated, taking 44 seconds. Adding the 98 seconds of interpolation time for the initial set of 9442 trajectories, the total computation time is approx. 2 minutes and 30 seconds, which is relatively high compared with the 3 minutes for the over 9000 trajectories of the first experiment.

Five clusters are identified with 25% of outliers. We can see the five centroids and the number of flights per cluster in

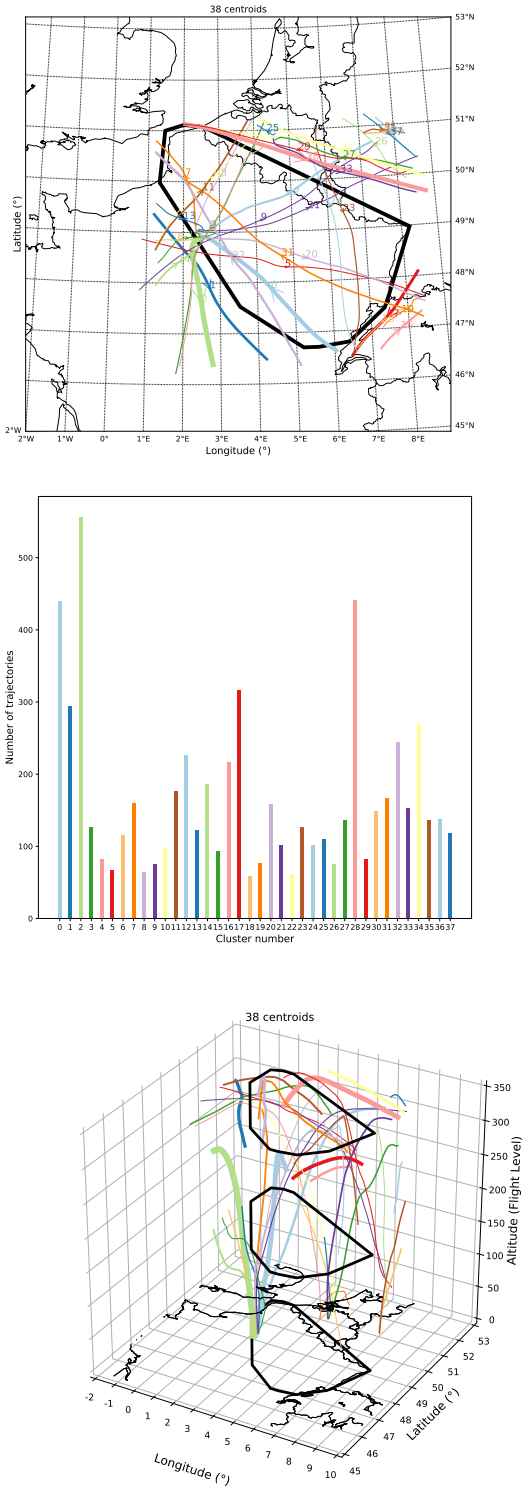


Figure 5. ED-based clustering results

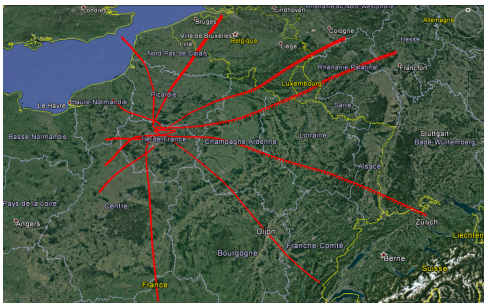


Figure 6. ED-based clustering - Centroids for Paris flows

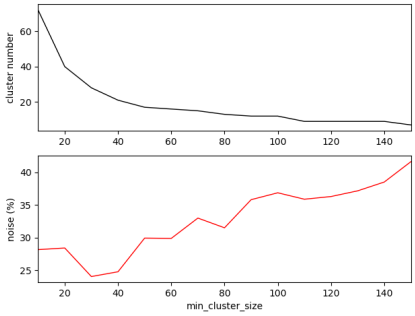


Figure 7. Influence of the cluster size on the number of clusters and percentage of outliers

Figure 8. It has to be noted that with SSPD a cluster can have trajectories with opposite directions because of the already-mentioned limitation of this distance and the fact that the clustering is performed in 2D. Therefore, the direction displayed for the cluster centroids may be the opposite direction of some of the trajectories in the cluster.

In Figure 9, we superimpose the identified clusters centroids in red over the air routes in green to make sure that both are consistent in terms at least of geographical location. For instance, cluster number 2 is a northwards flow located in the east of Paris matching perfectly with air route UM733.

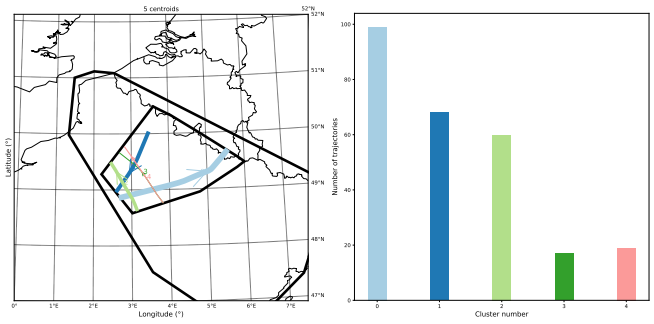


Figure 8. SSPD-based clustering results for LFEEXR sector

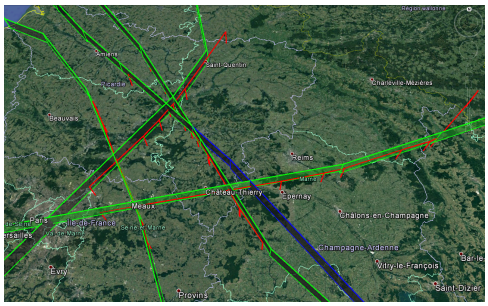


Figure 9. Matching cluster centroids (in red) and routes (in green and blue) in LFEEXR sector

TABLE II
ED VS SSPD RESULTS

	ED	SSPD
Number of trajectories after filtering	6133	6133
RDP threshold	None	300
Number of clusters	29	20
Noise/Outliers (%)	34	43
Execution time (sec)	306	89463

C. ED versus SSPD comparison

The last experiment consists of comparing both clustering methods by applying them to the same use case. The goal is to identify the major flows in 2D (minimum of 50 flights per day) in Reims ACC between FL300 and FL500. Clustering is performed in 2D because it is more convenient for visual verification, as controllers are used to a 2D view of the traffic. Parameters are shown in the third column of Table I and the results in Table II.

There is a significant difference in time performance between the two methods, with ED-based clustering being almost 300 times faster than SSPD-based clustering (about 5 minutes versus more than 24 hours). Additionally, ED-based clustering produces about 10% less outliers, which can be explained by the fact that SSPD better takes into account the differences in shape and physical distance between trajectories.

As for the quality of the clustering, ED performs worse in some cases like in cluster number 8 (see Figure 10). Even if the trajectories in this cluster are all southbound, it would be more appropriate to separate them into different clusters to better account for the diversity of headings. However, the ED distance is not sensitive enough to operate this separation and the only alternative would have been to decrease the minimum number of trajectories per cluster, which would result in an even a greater number of clusters. With SSPD, we obtain fewer and more homogeneous clusters, but the number of outliers (43 %) is considerable.

In order to further analyse the identified clusters, Table III gives the statistics for two of the flows identified by the SSPD method. Thus, 5% of the flights in cluster 7 follow the Paris Charles-de-Gaulle (LFPG) - Heathrow airport (EGLL) route, whereas 39% of the flights in cluster 8 follow the Nice (LFMN) - Paris Orly (LFPO) route.

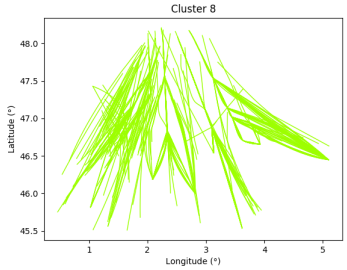


Figure 10. Cluster number 8 generated by the ED-based method

TABLE III
SSPD CLUSTERING STATISTICS EXAMPLE

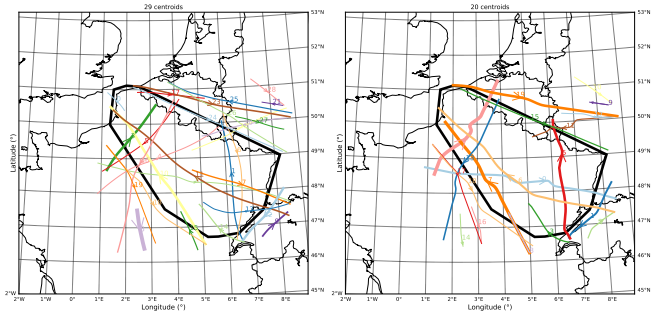
Cluster	7	8
Avg. Heading (°)	325	336
Avg. Alt (ft)	36199	32753
Avg. Length (NM)	255	75
Flights/hour	14.7	2.5
Main orig/dest	LFPG-EGLL	LFMN-LFPO
	5% (17)	39% (24)
Main origin	LFPG	LFMN
	16% (56)	43% (26)
Main destination	EGKK	LFPO
	17% (59)	89% (54)
Main A/C type	A319	A320
	22% (76)	62% (38)

It may not at first be evident as to why cluster 8 overlaps cluster 7, rather than being merged into one single cluster. In fact both clusters contain a majority of northbound flights from south-east France and Italy. However, most of the flights in cluster 8 are for Paris (89% LFPO), whereas in cluster 7 the main destination is London-Gatwick (17% EGKK), therefore there are two distinct routes that are represented by these two clusters.

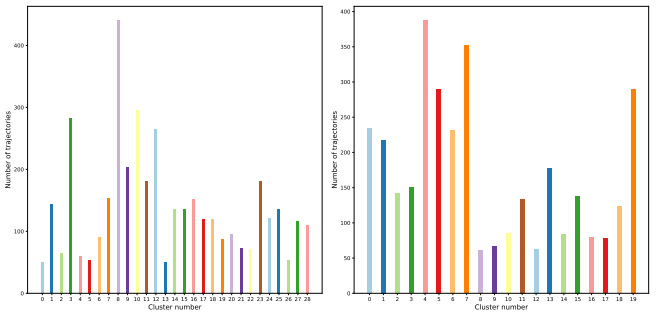
Figure 11 displays the centroids and their intensities (number of trajectories) to compare the results of both methods. It is not difficult to match the major flows, e.g. ED flows 3 (green), 10 (pale yellow), 1 (blue) correspond to SSPD flows 4 (salmon), 7 (orange), 5 (red). However, except for cases like the anomalous cluster ED 8 and the corresponding SSPD 14, SSPD clusters have a higher number of trajectories as this distance is unable to discriminate similar trajectories with opposite directions. For instance, we can observe that clusters 17 (red), 23 (brown), 25 (blue) identified by the ED method have been replaced by the single and bigger SSPD cluster 19 (dark orange).

D. Verification with planned trajectories

All of the previous experiments are based on the executed trajectories only (M3 in DDR2). However, the difference between the planned and executed traffic may be significant for days with an important volume of traffic as in our case, so it may be interesting to check how well the clustering



(a) ED (b) SSPD



(c) ED (d) SSPD

Figure 11. ED versus SSPD clustering results

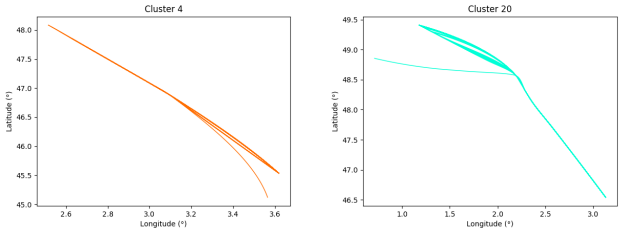


Figure 12. Clusters 4 and 20 generated with ED-based clustering

generated from the planned trajectories (M1 in DDR2) match the ATS Route Network (ARN). With the executed traffic, we have already seen that some of the clusters correspond to air routes in a previous experiment (see Figure 9). In the case of the planned trajectories, which are based on ARN and have not been modified by ATC in the tactical phase, the matching should be even more evident if our clusters are consistent with the operational reality.

We use the same parameters as those for ED (first column in Table I), i.e. with a minimum number of trajectories equal to 50. Even with the ED method, some of the obtained clusters are quite accurate e.g. clusters 4 and 20 in Figure 12.

These two clusters have 74 and 84 trajectories respectively. In each cluster, all trajectories are almost completely overlapping since no deviations from the flight plans were introduced by ATC. In Figure 13 it can be observed that sections of the published air routes UM133 and UM728 over Reims ACC (in green) match perfectly with the two clusters (in black). More

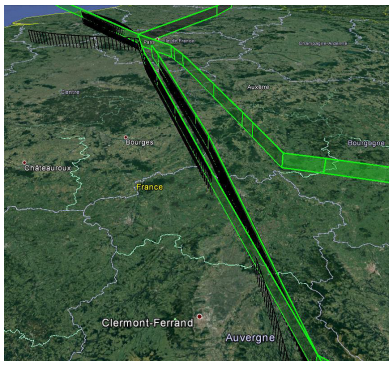


Figure 13. Comparison between routes UM133 and UM728 (in green) and clusters 4 and 20 generated by the ED-based method (in black)

generally, most of the clusters correspond to sections or links between sections of air routes. It is also interesting to note that even though the two routes are geographically close, the clustering is accurate enough to properly identify two separate flows.

IV. Conclusion

In this article, we have presented two methods based on the HDBSCAN clustering algorithm to identify air traffic flows from a set of trajectories. The choice of the method to use depends on the volume of trajectories to be processed and the desired accuracy. For a precise clustering, over a limited area, such a sector or small family of sectors, SSPD-based clustering is the best-adapted, except if we want to make sure that similar 2D flows in opposite directions are properly separated. Otherwise, ED-based clustering is probably the best option as SSPD may be prohibitively slow for some applications unless parallel computing for SSPD is implemented.

Our results have shown that we can match the obtained clusters to existing operational air routes. For the planned trajectories, clusters fit the route structure over Reims as expected, which reinforces our confidence in the framework results. On the other hand, outliers are in general quite high and further analysis is needed to characterise them. The best way to validate the framework would be to use it in a real application and have the operational experts (Flow Management Position or controllers) check that the identified flows make sense.

In particular, we are considering using the framework within the scope of the SESAR PJ08 project (Advanced Airspace Management) [21], where sector configurations could be optimised by minimizing flow cuts. In addition, we need to add further analytic capabilities in order to better understand the contribution of each flow, as well as that of the outliers, to the traffic complexity in a sector.

We would like to extend the framework by adding other clustering algorithms. A segment approach such as the one in TRACLUS seems particularly promising, where a trajectory could potentially be associated not only to a single cluster, but to a sequence of clusters, to better explain the different flight phases. Thus, in the en-route phase, the trajectories sharing the same air routes could have their en-route segments clustered, independently of whether or not they share the same departure/arrival procedures.

Acknowledgment

The authors would like to thank Sébastien Aubry, Judicaël Bedouet, Thomas Dubot, Antoine Joulia and Xavier Olive from ONERA and Pascal Lezaud from ENAC for their contributions to the present work.

References

[1] S. J. Undertaking, "European ATM master plan," Tech. Rep., 2015.
[2] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
[3] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," in *ACM Sigmod Record*, vol. 25, no. 2. ACM, 1996, pp. 103–114.
[4] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[5] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.
[6] M. C. R. Murça, R. DeLaura, R. Hansman, R. Jordan, T. Reynolds, and H. Balakrishnan, "Trajectory clustering and classification for characterization of air traffic flows," *AIAA Aviation*, 2016.
[7] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1511–1524, 2011.
[8] 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*. ACM, 2007, pp. 593–604.
[9] A. T. Nguyen, "Identification of air traffic flow segments via incremental deterministic annealing clustering," Ph.D. dissertation, 2012.
[10] M. Enriquez, "Identifying temporally persistent flows in the terminal airspace via spectral clustering," in *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)*, 2013.
[11] M. K. El Mahrsi and F. Rossi, "Graph-based approaches to clustering network-constrained trajectory data," in *International Workshop on New Frontiers in Mining Complex Patterns*. Springer, 2012, pp. 124–137.
[12] S. Puechmorel and F. Nicol, "Entropy minimizing curves with application to flight path design and clustering," *Entropy*, vol. 18, no. 9, p. 337, 2016.
[13] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2013, pp. 160–172.
[14] B. Guillouet, "Apprentissage statistique: application au trafic routier à partir de données structurées et aux données massives," Ph.D. dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2016.
[15] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, mar 2017. [Online]. Available: https://doi.org/10.21105/joss.00205
[16] Eurocontrol. DDR repository. (2017). [Online]. Available: http://www.eurocontrol.int/ddr
[17] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer graphics and image processing*, vol. 1, no. 3, pp. 244–256, 1972.
[18] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
[19] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238–246, 1980.
[20] D. Kahaner, C. Moler, and S. Nash, "Numerical methods and software," *Englewood Cliffs: Prentice Hall, 1989*, 1989.
[21] "SESAR Solution 08.01 SPR-INTEROP/OSED for V2 - Part1, Edition 00.01.00, D2.1.020," Tech. Rep., 2017.