D2.2 VISUAL INTERFACE FOR ALGORITHMS ANALYSIS

DART

Grant:	699299
Call:	ER-1-2015
Topic:	Data Science in ATM
Consortium coordinator:	University of Piraeus Research Center
Edition date:	29 August 2017
Edition:	[02.01.00]

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Natalia Andrienko, Gennady Andrienko, Georg Fuchs / FRHF	Project Members/Researchers	16.05.2017

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
George Vouros/UPRC	Project Coordinator	06.06.2017
David Scarlatti/BR&T-E	Project Member	06.06.2017
Jose Manuel Cordero/CRIDA	Project Member	06.06.2017

Approved for submission to the SJU By — Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
George Vouros/UPRC	Project Coordinator	16.06.2017
David Scarlatti/BR&T-E	Project Member	16.06.2017
Georg Fuchs/FRHF	Project Member	16.06.2017
Jose Manuel Cordero/CRIDA	Project Member	16.06.2017

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Alessandro Prister	SJU Programme Manager	02.08.2017
Andreas Hasselberg	SJU Expert	02.08.2017

Revised by

Name/Beneficiary	Position/Title	Date
Natalia Andrienko / FRHF	Project Member/Researcher	29.08.2017

Document History

Edition	Date	Status	Author	Justification
01.00.00	16.05.2017	First Draft	Natalia Andrienko, Gennady Andrienko	Document initiation
01.01.00	16.06.2017	Revision	Georg Fuchs	Completed Sect. 1
01.01.00	02.08.2017	Evaluation by SJU	Alessandro Prister, Andreas Hasselberg	Evaluated
02.01.00	29.08.1017	Revision	Natalia Andrienko	Revised according to the evaluation remarks

DART

DATA DRIVEN AIRCRAFT TRAJECTORY PREDICTION RESEARCH

This document is part of a project that has received funding from the SESAR Joint Undertaking under Grant Agreement No 699299 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document describes visualization and interaction techniques and interfaces supporting the exploration and evaluation of results of trajectory prediction algorithms, particularly, comparison of predicted trajectories to real ones and comparison of predictions obtained with different algorithms or different parameter settings. The described techniques support pairwise comparison of trajectories by (a) finding and linking matching points, (b) generating and visualizing attributes representing the spatial and temporal distances between the matched points and differences in their attribute values, and (c) enabling interactive querying and filtering of the results of point matching.

Table of Contents

Abstract 4

Table of Contents *5*

List of Figures *6*

1 Introduction *7*

1.1 Purpose and Scope 7

1.2 Intended readership 8

1.3 Acronyms 8

1.4 Relation to other Work Packages and Deliverables 8

1.5 Software used 8

2 Basic techniques for visualization and exploration of trajectory data *10*

2.1 Map-based visualization 10

2.2 Space-time cube 14

2.3 Globe view of flight trajectories 17

2.4 Time line and time series views 19

2.5 Interaction techniques 22

3 Support of pairwise comparison of trajectories *25*

3.1 Problem statement and background 25

3.2 Computation of pairwise differences 26

3.3 Interactive visually supported exploration of pairwise differences between trajectories 30

4 Support of exploration of multiple comparison results *37*

4.1 Statistics of point-wise differences 37

4.2 Exploration of the spatial and temporal distribution of deviations 41

4.3 Conclusion 44

5 References *46*

List of Figures

Figure 1 – Representation of a single trajectory on a map 10
Figure 2 – A set of flight trajectories is shown on a map, each trajectory being drawn with 90% transparency. 11
Figure 3 – Flight trajectories are coloured according to the values of the attribute ‘airline’. 11
Figure 4 – Line widths represent the difference of the actual flight duration with respect to the flight plan. 12
Figure 5 – Flight subsets chosen based on delay duration. 12
Figure 6 – Representation of flight level change rates. 13
Figure 7 – Representation of the change rates of the ground speed and flight level. 13
Figure 8 – A space-time cube display. 14
Figure 9 – Examples of planned and actual trajectories shown in a space-time cube. 15
Figure 10 – A tilted STC view. 16
Figure 11 – An STC with a movable plane. 16
Figure 12 – A tilted STC view with a movable plane. 17
Figure 13 – A globe view of a set of flights. 18
Figure 14 – Another perspective into the globe view. 19
Figure 15 – A time line display. 20
Figure 16 – Dynamic linking between a map display and a time line display. 20
Figure 17 – A time graph display. 21
Figure 18 – A time graph display after applying time transformation. 21
Figure 19 – Interactive filtering of trajectory segments. 23
Figure 20 – Effect of interactive filtering of trajectory segments. 23
Figure 21 – Trajectories with horizontal flight segments hidden. 24
Figure 22 – Examples of results of point matching. 30
Figure 23 – Interactive access to attributes of trajectories and their points. 31
Figure 24 – Interactive access to attributes of the links between the matched points. 31
Figure 25 – Temporal differences shown in a space-time cube. 32
Figure 26 – Comparison of the temporal developments of planned and actual flights using a time graph. 33
Figure 27 – Comparison of planned and actual flights using a map. 33
Figure 28 – Comparison of flight levels using a 3D display. 36
Figure 29 – Exploration of the proportions of matched points. 38
Figure 30 – Exploration of the temporal gaps between matched points. 39
Figure 31 – Exploration of the spatial distances between matched points. 40
Figure 32 – Exploration of temporal distribution of matched points. 42
Figure 33 – Exploration of the spatial distribution of matched points. 44

1 Introduction

1.1 Purpose and Scope

The purpose of this document is to describe the visual and interactive techniques and interfaces designed to support pairwise comparison of corresponding trajectories, such as predicted trajectories against real, flight plans against actual flights, and predictions obtained using different algorithms or different parameter settings. The purpose of designing these techniques is to support the development and evaluation of the algorithm(s) for single trajectory prediction in work package WP2, as well as studying the sensitivity of the algorithm(s) to parameter settings and finding suitable values of algorithm parameters.

The document describes three groups of techniques:

1. Pre-existing state of the art techniques that were chosen for use in DART due to their suitability for visual exploration of aircraft trajectories.
2. New techniques that were developed within DART with the purpose to support comparative analysis of aircraft trajectories.
3. Pre-existing techniques that are suitable for interactive visual exploration of the results of the new techniques.

Regarding the first group of techniques, which is described in chapter 2 of this document, the work in DART consisted of:

- Experiments on applying the techniques to DART data for exploring their suitability and effectiveness.
- Selection of useful techniques and appropriate visualization parameters.
- Creation of examples showing the use of the techniques and demonstrating their usefulness.

The second group of techniques is described in chapter 3 of this document. The techniques support pairwise comparison of aircraft trajectories by finding and linking corresponding points and showing the matches, mismatches, and differences in space, time, and movement attributes. The work that has been done includes:

- Development of a point matching algorithm (section 3.1).
- Development of new interactive visualization techniques to show links and differences between matched points (section 3.3).

In chapter 4, we address the problem of visual exploration of matching results for large numbers of trajectories. This problem requires aggregation of the results and visualization of the aggregates, which can be done using pre-existing techniques, such as histograms and density maps. The work in DART consisted of:

- Finding appropriate ways to aggregate the results of the trajectory matching.
- Selection of suitable visualization and interaction techniques to visualize and explore the aggregated results.
- Creation of examples showing the use of the techniques and demonstrating their usefulness.

Although the techniques described in chapter 3 were developed specifically for DART, their applicability is not limited to DART data only. In fact, the techniques can be applied to any pairs of spatially similar trajectories. Hence, the results of this development can be reused (possibly, after adaptation) in other projects where pairwise comparison of trajectories is needed.

1.2 Intended readership

This document is intended to be used by DART consortium members.

Dissemination Level: Confidential, only for members of the consortium (including the Commission Services)

1.3 Acronyms

Term	Definition
STC	Space-Time Cube
VA	Visual Analytics

1.4 Relation to other Work Packages and Deliverables

The work described in this deliverable relates directly to further work carried out in work package 2 as described in D2.1, Data-driven Trajectory Prediction Algorithms, and work carried out in work package 3 as described in D3.1, Collaborative Trajectory Prediction. Visualization and visual exploration methods detailed in this document have been applied to assess raw data available in DART, as well as to explore model results such as predicted trajectories vs. actual flights.

1.5 Software used

The techniques described in this document are implemented in a software system V-Analytics developed within Fraunhofer IAIS. The software has been evolutionary developed, partly within a series of EU-funded projects. New functions and techniques are continually included in the system according to the needs of different projects.

Trajectory data can be loaded in the system from text files in CSV format or from a database. The data must include the following mandatory fields:

- Identifier of a moving object (e.g., an airplane)
- Time stamp
- Spatial coordinates of the moving object at the specified time

The data may also include additional fields with any attributes referring to the times and spatial positions of the moving objects. In particular, aviation data are expected to include the flight altitudes and/or flight levels. A trajectory may consist of any number of points. The system does not do pre-processing of the data, such as interpolation and resampling. If some pre-processing is needed for analysis, it can be done using external tools, and the results can be loaded in V-Analytics for visualization and exploration.

2 Basic techniques for visualization and exploration of trajectory data

2.1 Map-based visualization

Trajectories are visualized on a map by lines connecting consecutive points. Figure 1 shows an example of a single flight trajectory from Barcelona to Madrid, and Figure 2 shows a set of trajectories of 3,410 flights from Barcelona to Madrid conducted during the period from January to May 2016.

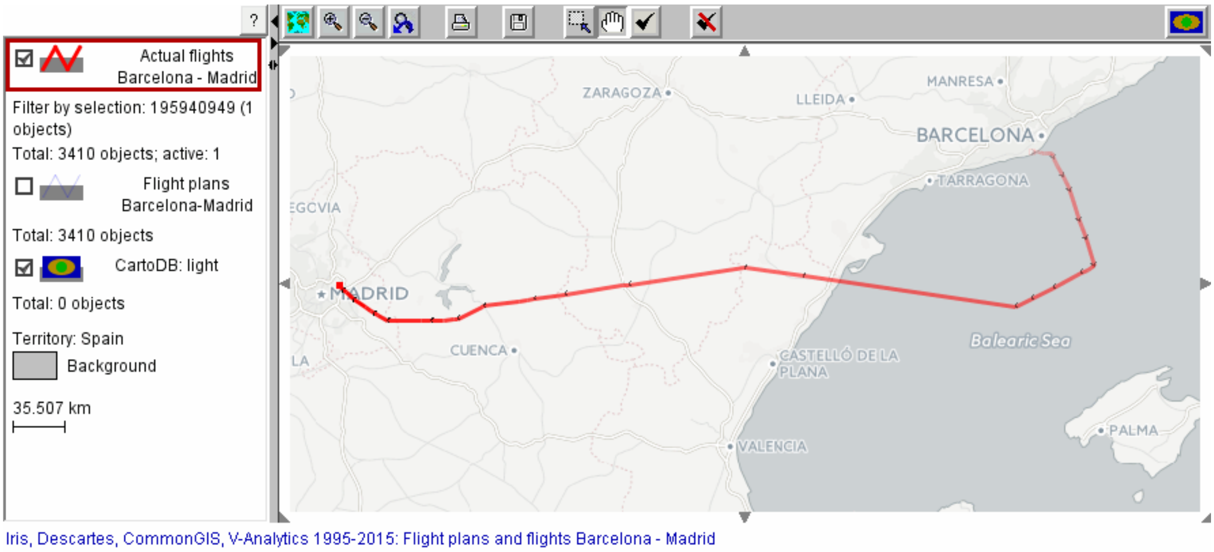


Figure 1 – Representation of a single trajectory on a map

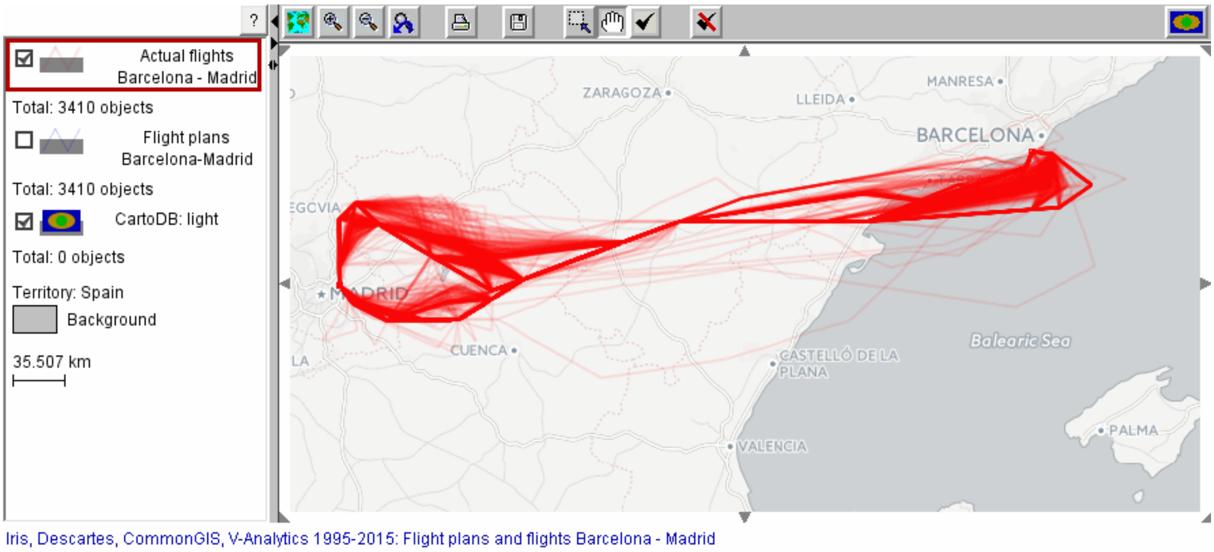
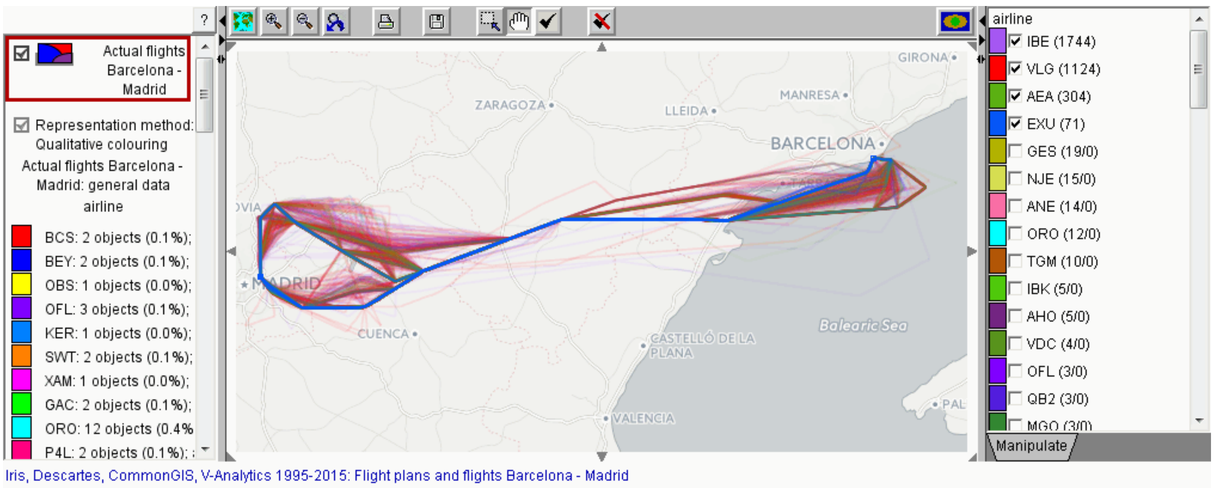


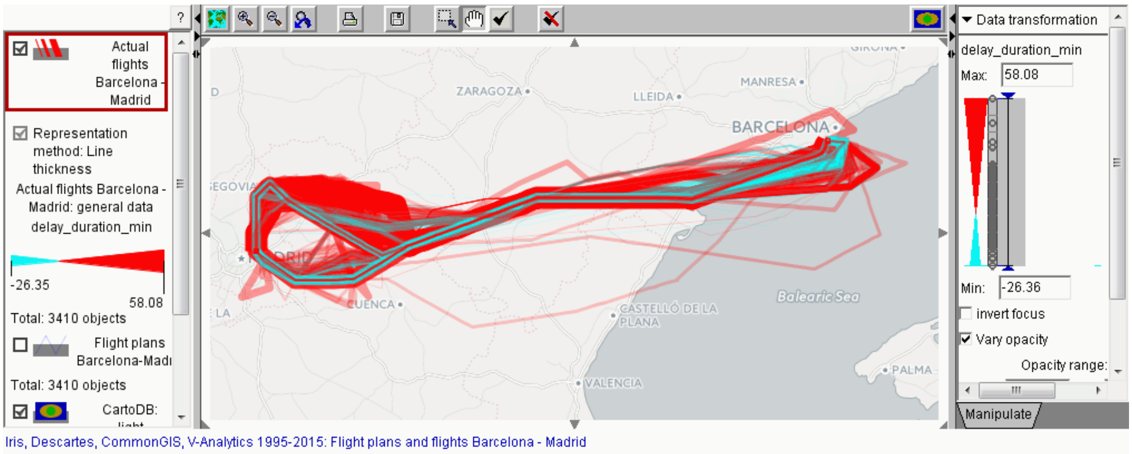
Figure 2 – A set of flight trajectories is shown on a map, each trajectory being drawn with 90% transparency.

Trajectories may have various attributes, and the values of these attributes may be represented on a map by colours or shades in which the trajectory lines are painted or by the line widths. As an example, the colours of the trajectory lines in Fig. 3 represent different airlines that conducted the flights. In Fig. 4, the differences of the actual flight duration with respect to the plan are represented by combinations of colour hues and line widths. The red and cyan colour hues represent positive and negative differences, respectively, while the line widths are proportional to the absolute values of the differences. In Fig. 5, top and bottom, the flights with the positive and negative duration differences are shown separately.



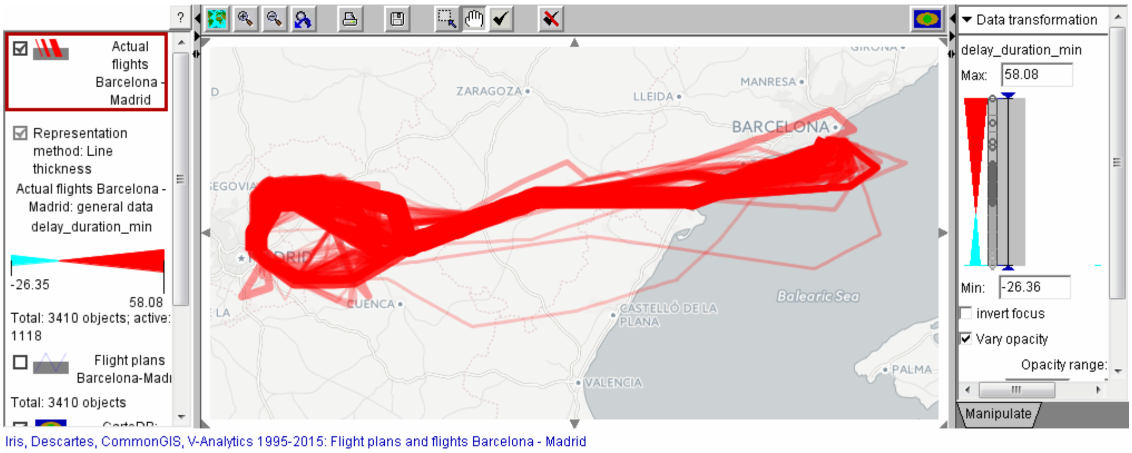
The panel on the right contains interactive controls for filtering of the trajectories based on the attribute values.

Figure 3 – Flight trajectories are coloured according to the values of the attribute 'airline'.

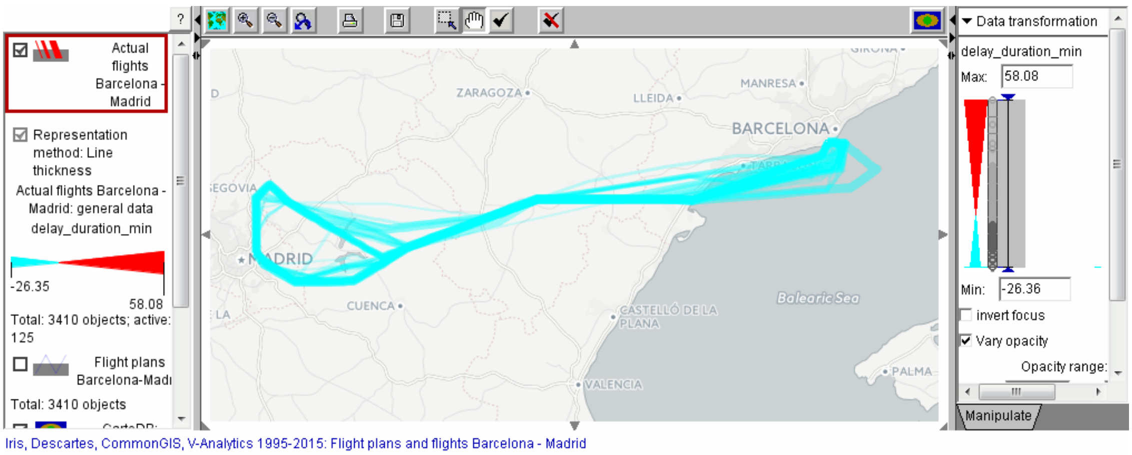


Positive differences are shown in red and negative in cyan.

Figure 4 – Line widths represent the difference of the actual flight duration with respect to the flight plan.



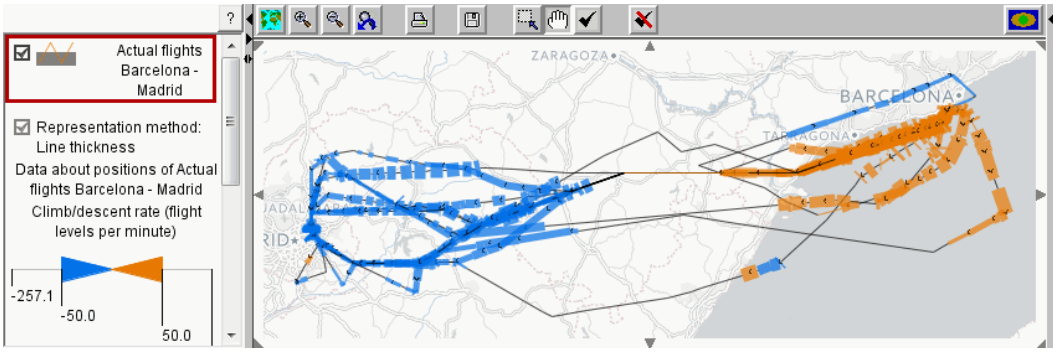
The subset of flights that took at least 5 minutes longer than planned.



The subset of flights that took at least 5 minutes less than planned.

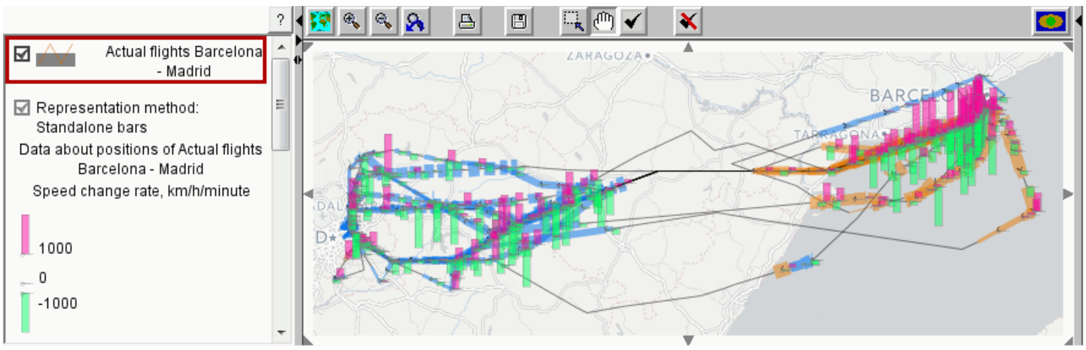
Figure 5 – Flight subsets chosen based on delay duration.

Besides attributes referring to whole trajectories, there may be attributes associated with the points or segments of trajectories. These are called ***positional attributes***. Values of positional attributes can be visualized on a map by colours, shades, or widths of trajectory segments, i.e., the lines connecting the points. They can also be represented by symbols or diagrams attached to the points of the trajectories. Examples are shown in the following figures. In Fig. 6, colours and widths of line segments represent values of attribute 'Climb/descent rate'. Dark orange corresponds to flight level increase (climb), and blue to flight level decrease (descent). The line widths are proportional to the rates of the changes (flight level change per minute). Figure 7 shows the same information as in Fig. 6 and, additionally, the values of the attribute 'Ground speed change rate'. These values are represented by vertical bars positioned at the points of the trajectories. The bars in pink and green represent speed increase and decrease, respectively, while the bar heights are proportional to the rates of change.



Colours and widths of line segments represent climb or descent rates, i.e., flight level changes per minute. Dark orange corresponds to flight level increase (climb) and blue to decrease (descent). The line widths are proportional to the flight change rates.

Figure 6 – Representation of flight level change rates.

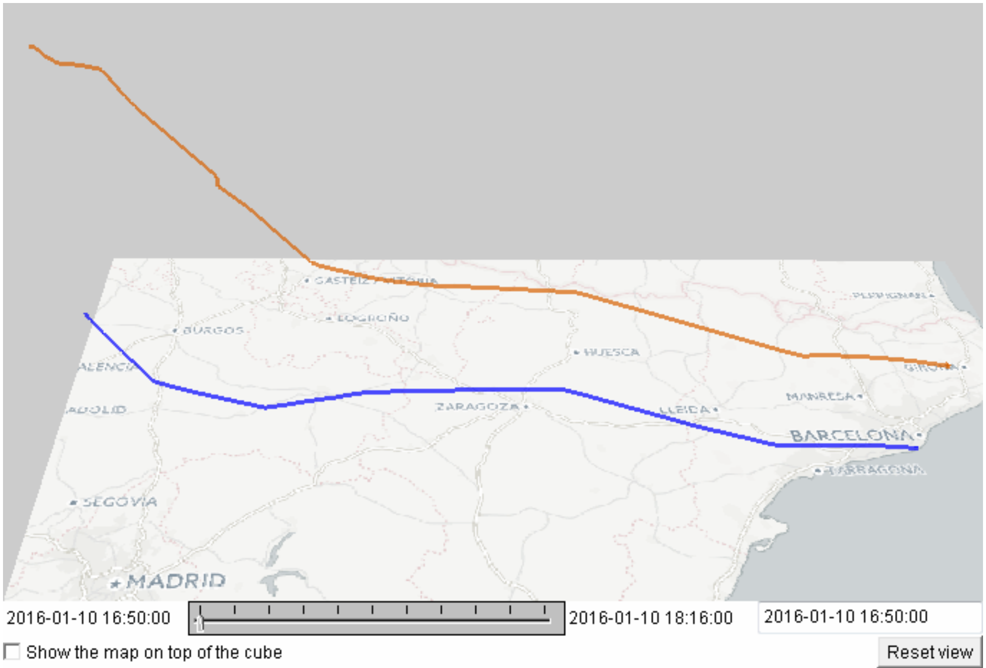


Vertical bars represent change rates of the ground speed (i.e., acceleration or deceleration) in the trajectory points, measured in km/h per minute. Pink colour corresponds to speed increase (acceleration) and green to speed decrease (deceleration). The bar heights are proportional to the change rates.

Figure 7 – Representation of the change rates of the ground speed and flight level.

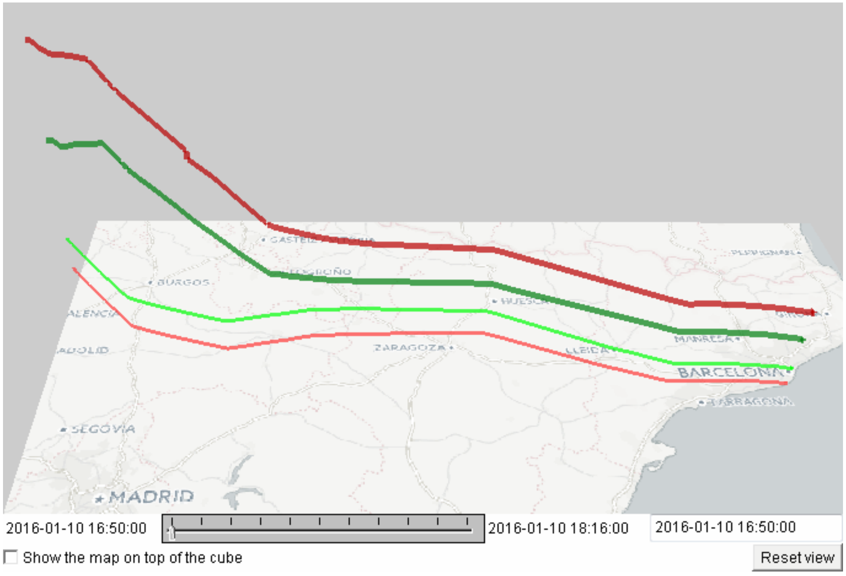
2.2 Space-time cube

A space-time cube (STC) is a display representing two-dimensional (2D) space and time as an integrated space-time continuum. The display is a perspective view of a 3D scene where two display dimensions represent the two spatial dimensions and one dimension represents time. In this display, trajectories are represented by polygonal lines obtained by connecting consecutive points, the latter being placed according to the spatial locations and times of the corresponding positions in the trajectories. Examples are provided in Figs. 8 and 9 below. Like in a map display, attributes of trajectories can be represented by line colouring, shading, or widths. The STC display may be convenient for exploration of spatio-temporal characteristics of a single trajectory or a few trajectories within a short time interval. With a larger number of trajectories and/or longer time interval, the display may become cluttered and insufficiently expressive.



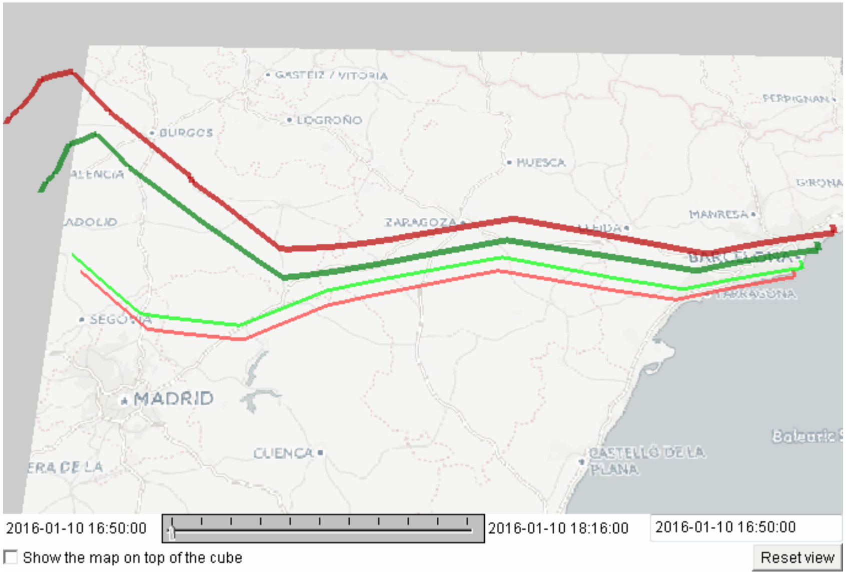
The base of the cube represents two-dimensional geographic space and the vertical dimension represents time. The time axis is oriented upwards. The blue and orange lines in the display represent trajectories of a flight plan and a corresponding actual flight, respectively. The vertical gap between the lines shows the delay of the actual flight with respect to the plan. Besides, the lines differ in the shapes of their ending parts, which means that the actual flight route differed from what was planned.

Figure 8 – A space-time cube display.



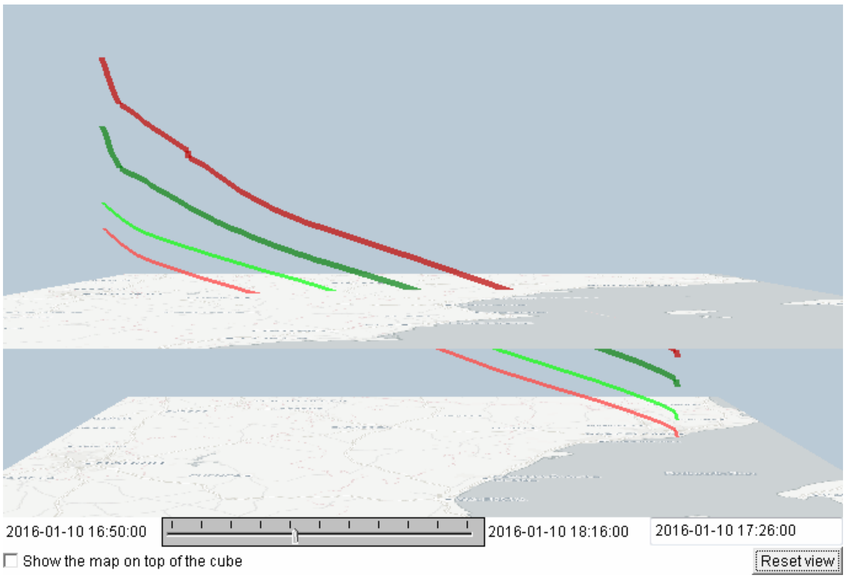
For a selected time interval, the STC shows the trajectories of the flights that were conducted and the trajectories of the corresponding flight plans. The red and green colours correspond to two distinct airlines, VLG and IBE. The thicker and darker lines represent the actual flights and the lighter and thinner lines represent the corresponding flight plans. It is seen that the IBE flight (green) was planned to be conducted later than the VLG flight but it actually happened earlier. Both flights had delays with respect to the plans, but the VLG flight delay was much longer.

Figure 9 – Examples of planned and actual trajectories shown in a space-time cube.



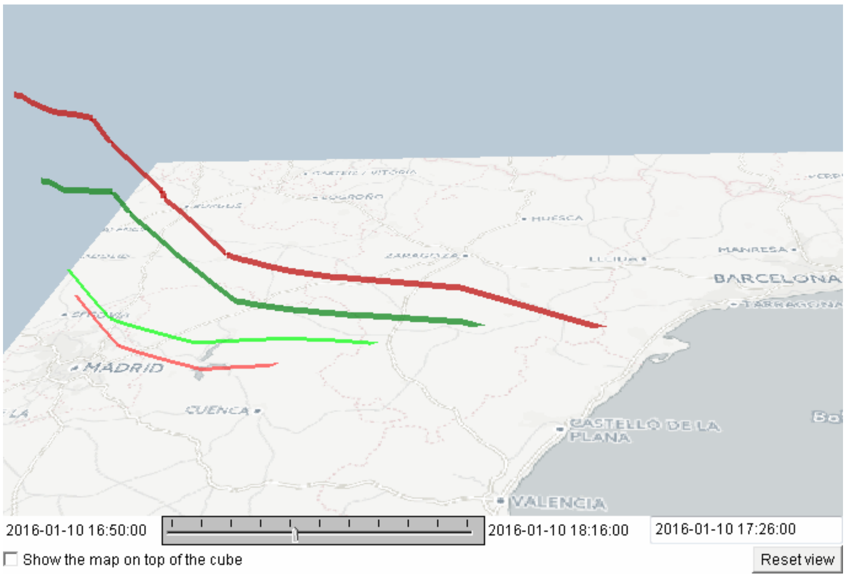
An STC view can be tilted to see better the differences between the routes. Now it may be seen that the ending parts of both actual flights significantly differed from the planned routes.

Figure 10 – A tilted STC view.



A movable plane in the STC corresponds to a particular time moment within the interval represented in the display. It separates the parts of the trajectories preceding the chosen moment from the following parts.

Figure 11 – An STC with a movable plane.

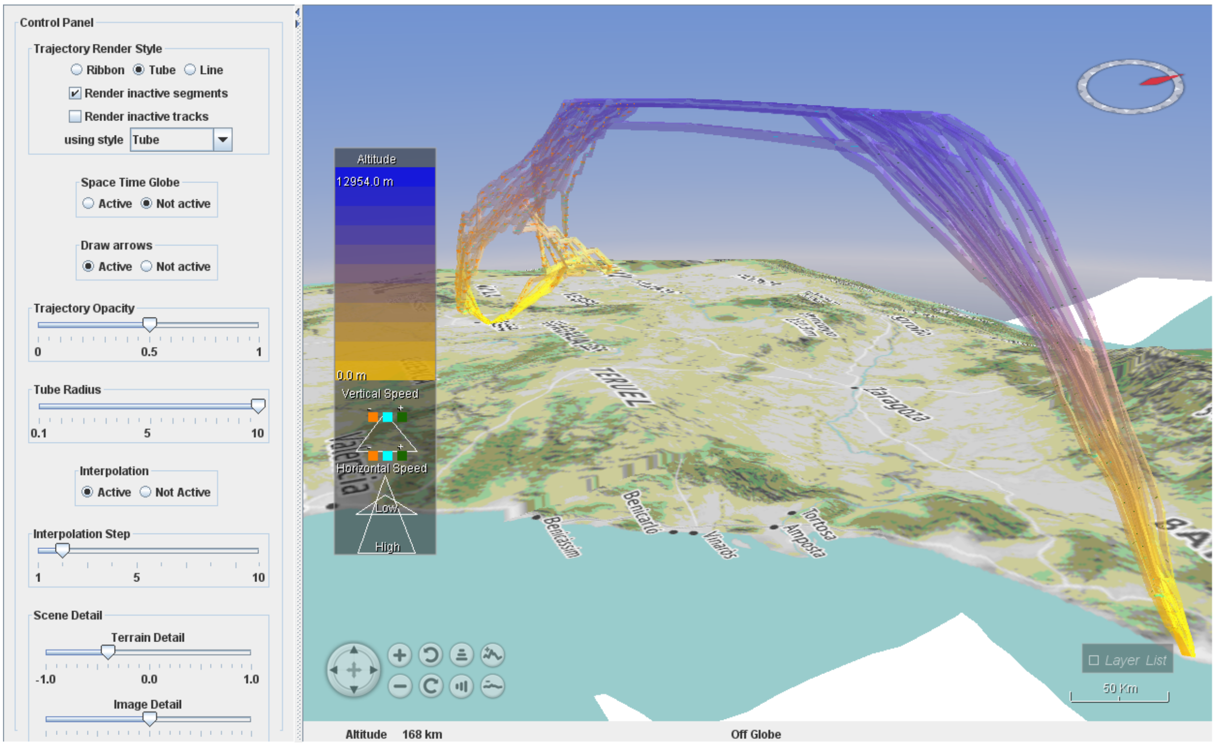


The STC view with a movable plane shown in Fig. 11 has been tilted to see more clearly the planned and actual positions of the planes at the chosen time moment and the remaining travel paths.

Figure 12 – A tilted STC view with a movable plane.

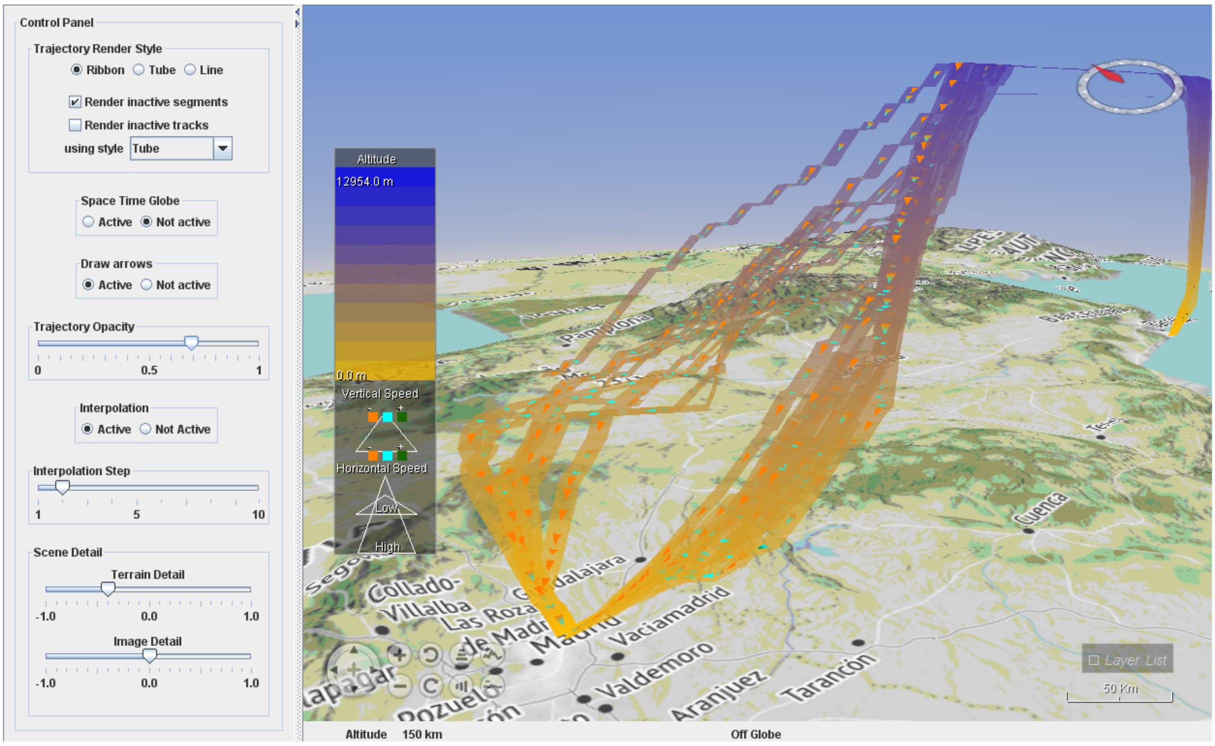
2.3 Globe view of flight trajectories

Flight trajectories are assumed to have a positional attribute whose values reflect the altitudes or flight levels attained at the different positions in the trajectories. Such trajectories can be visualized in a 3D display, called globe view, where the vertical dimension represents the altitudes or flight levels. Trajectories can be represented by lines, tubes (Fig. 13), or ribbons (Fig. 14). Colour variation along trajectories can represent positional attribute values; thus, in Figs. 13 and 14, the colour variation from yellow to dark blue represents the altitude variation from the lowest to the highest. An additional positional attribute, as the vertical speed in Fig. 14, can be represented by triangular symbols drawn on top of ribbons.



Flights from Barcelona to Madrid conducted during one day are represented by tubes. The colour variation represents the attained altitudes. The globe is rotated to focus on the starting parts of the flights.

Figure 13 – A globe view of a set of flights.

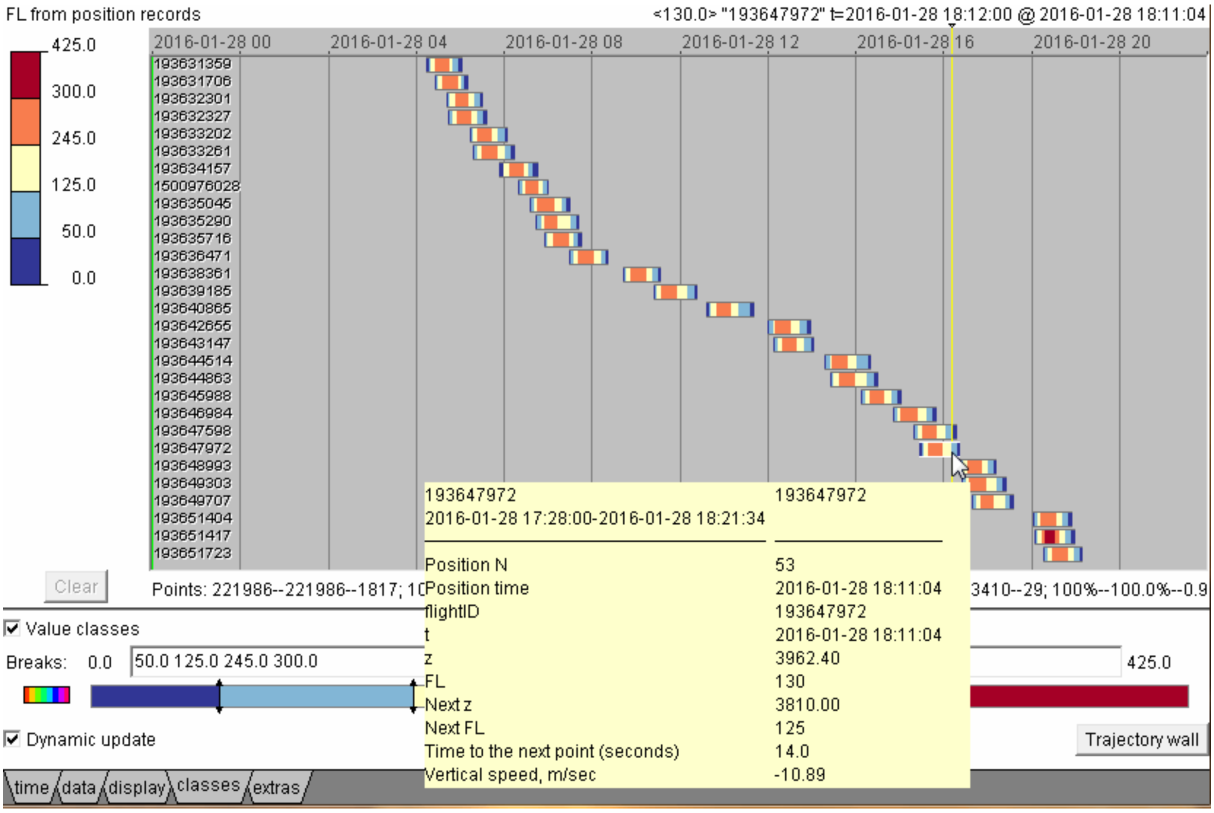


The same set of flights as in Fig. 13 is shown from another perspective, so that the ending parts of the trajectories were better visible. The trajectories are represented by ribbons. The triangular symbols show the vertical speed values (orange corresponds to negative values, i.e., decreasing altitudes).

Figure 14 – Another perspective into the globe view.

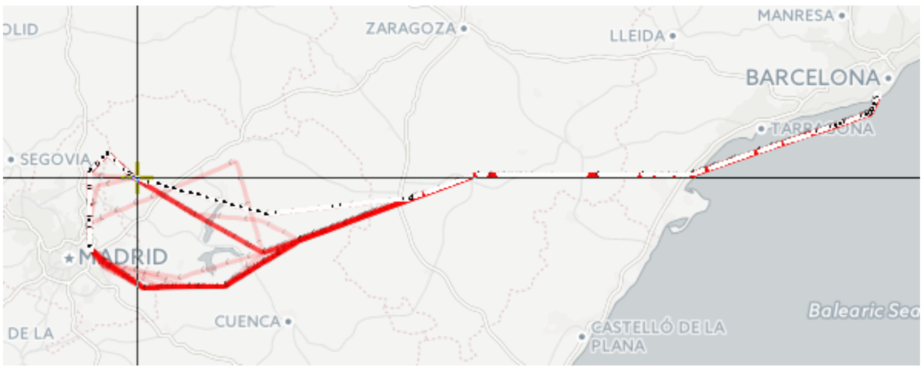
2.4 Time line and time series views

In a 2D temporal display, one of the dimensions (typically horizontal) represents time. In a time line display, the vertical dimension is used to represent trajectories by stacked horizontal bars, one below another. The horizontal positions and lengths of the bars correspond to the time intervals when the trajectories happened; hence, the bar lengths are proportional to the durations of the trajectories. Values of a selected positional attribute may be represented by colours of segments in which the bars are divided (Fig. 15). By moving the mouse cursor along a bar, the user obtains a popup window showing the values of positional attributes for the corresponding trajectory positions. Moreover, the position currently pointed at with the mouse cursor is marked in the map display by an intersection of horizontal and vertical lines (Fig. 16); hence, the user can see where in space this position is located. The link also works in the opposite direction, i.e., from the map to the time line view.



Trajectories are represented in a time line display by stacked horizontal bars. The horizontal positions and lengths of the bars correspond to the time intervals when the trajectories happened. The bars can be divided into segments coloured according to values of a selected positional attribute.

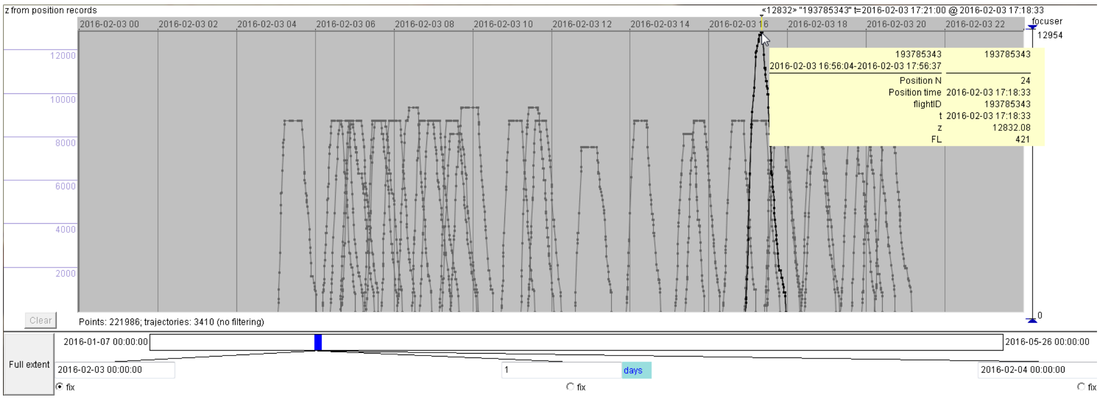
Figure 15 – A time line display.



The position pointed at with the mouse cursor in the time line display (Fig. 15) is marked in the map display by the intersection of horizontal and vertical lines.

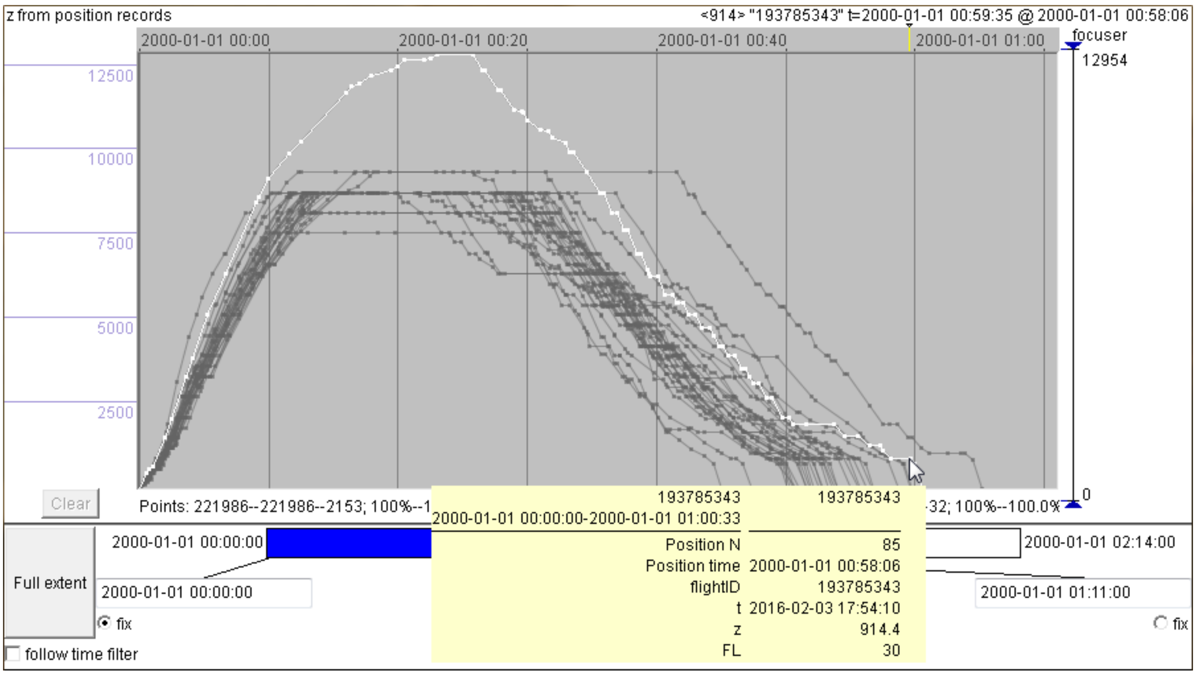
Figure 16 – Dynamic linking between a map display and a time line display.

In a time graph display (Figs. 17 and 18), the horizontal dimension represents time and the vertical dimension corresponds to the value range of a numeric positional attribute. Points of trajectories are represented by points in the graph positioned according to their times and the associated attribute values. The consecutive points are connected by lines. As an example, the graphs in Figs. 17 and 18 represent the altitude variation in flights conducted during one day. Like in a time line display (Fig. 15), points may be pointed at with the mouse to see their attribute values. The geographic positions of these points are marked on the map in the same way as in Fig. 16.



Time series of values of a numeric positional attribute associated with trajectory points are represented in a time graph display by polygonal lines. The line vertices correspond to the trajectory points. They are positioned horizontally according to the time references of the points and vertically according to values of the attribute.

Figure 17 – A time graph display.



Dynamic movement characteristics in multiple trajectories can be more conveniently compared after transforming the original times of the trajectory points to relative times with respect to the movement start or end. Here, the same data as in Fig. 17 are shown after transforming the times to relative with respect to the flight starts.

Figure 18 – A time graph display after applying time transformation.

To compare dynamic movement characteristics (i.e., temporal variation of positional attribute values) in multiple trajectories that happened at different times (and, possibly, during a long time period), it is convenient to have the lines representing the trajectories aligned in the time graph so that their starts or ends are shifted to the same horizontal position. This can be achieved by transforming the original times of the position records to relative times with respect to the movement start or end. Thus, Figure 18 enables convenient comparison of altitude dynamics in a subset of flights from Barcelona to Madrid conducted during one day. The times have been transformed to relative with respect to the flight starts. As a result, the polygonal lines within the graph are horizontally aligned to the same starting position. The same information is shown in Fig. 17, but there the lines are harder to compare. Besides, each line receives only a small part of the display widths; therefore, less detail is visible.

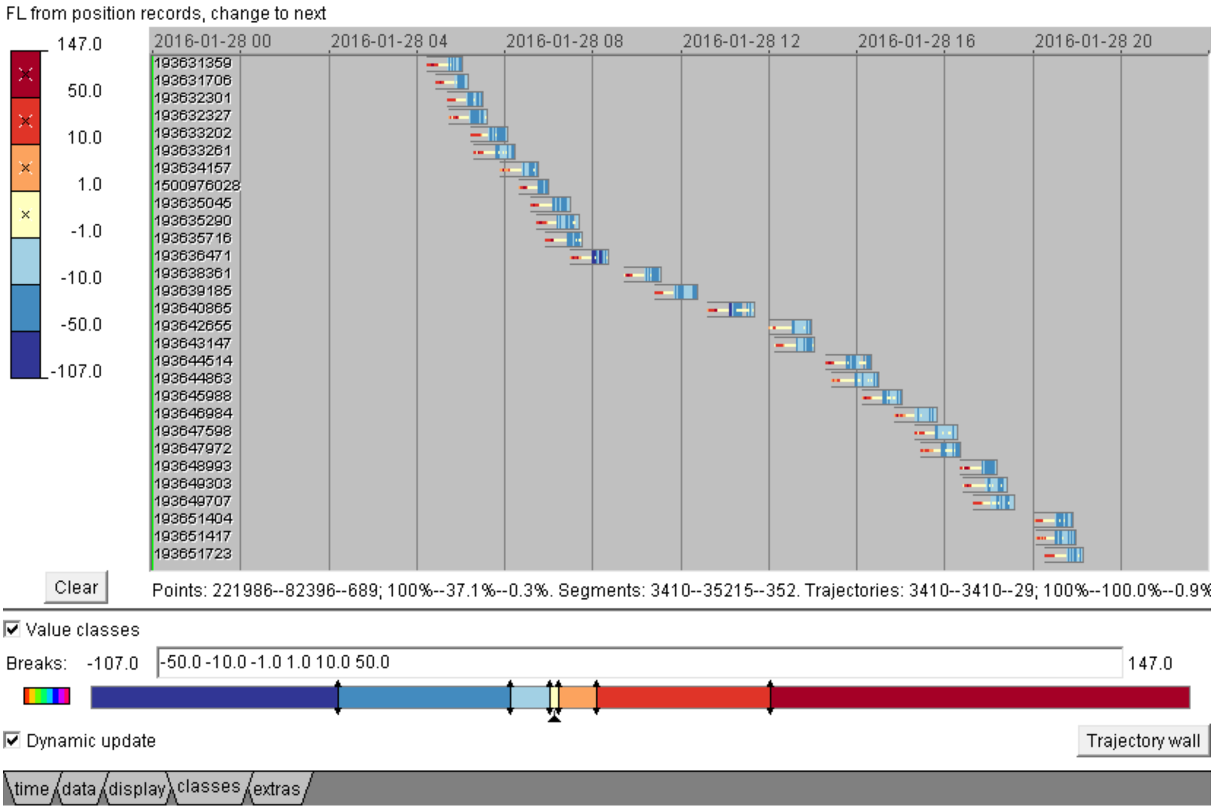
2.5 Interaction techniques

Some of the basic interaction techniques have been shown in the previous illustrations. Attribute values can be seen in a popup window when the mouse cursor points at a position where some data are represented in a display. Figures 15, 17, and 18 show examples with time line and time graph displays, but this technique works in a similar way in maps and other displays representing individual objects (not aggregated data). Different displays are linked through simultaneous highlighting of corresponding objects (e.g., visual objects representing trajectories) when the mouse cursor points at one or several objects in one of the displays. Highlighted objects are marked using white colour, as shown in Figs. 15-16 and 18. Mouse clicking on an object selects it, or deselects if it was selected previously. Selected objects are specially marked in all displays, usually in black colour (Fig. 17) and, in some displays, with the use of thick lines.

An important interactive operation is filtering, when the user interactively specifies what data items need to be shown. The remaining data are temporarily hidden. The query conditions can be specified in various ways and according to different criteria, including values of attributes, spatial locations, and temporal references. The types of filters that can be useful in exploration and analysis of movement data are described in the book by Andrienko et al. (2013).

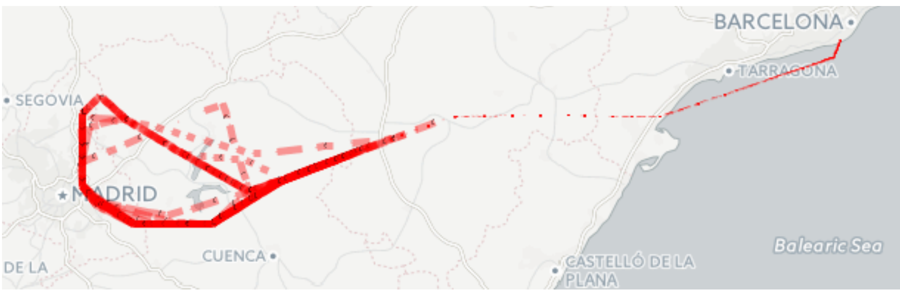
Examples of filtering occur in the previous illustrations. Thus, the single trajectory shown in Fig. 1 was first selected by mouse clicking in the map, and then the selection-based filter was applied. Similarly, several trajectories were selected for exploration in Figs. 6-8. In Fig. 3, the flights of four major airlines were selected by filtering based on values of the categorical attribute 'airline'. In Fig. 5, subsets of trajectories with positive and negative differences of the flight duration with respect to the plan were selected based on the values of the numeric attribute 'Flight duration difference'. In Figs. 9-16, time-based filtering was applied to select subsets of flights that occurred during particular time intervals. In Figs. 9-12, the interval length was about 1.5 hours, and in Figs. 13-16, it was one day.

In all these examples, filtering selected entire trajectories, but filtering can also be applied to points and segments of trajectories. An example of applying segment filter is shown in Figs. 19 and 20. In Fig. 19, the "active legend" on the left of a time line display has been used to filter out the trajectory segments where the flight level increases or is constant. In response, the map in Fig. 20 shows the trajectory segments where the flight level decreased by thick solid lines and the remaining segments by thin dotted lines (the user can also make these segments completely hidden). It is interesting that in some trajectories descending segments, which are visible, alternate with other segments that have been hidden by the filter. Figure 21 shows the result of changing the filter so that only the horizontal segments (i.e., where the flight level remained constant) are hidden. We can see that in the trajectories that we have noticed earlier the segments with the decreasing flight level alternate with horizontal segments, which corresponds to step-wise descent.



The filter, which is set using the active legend on the left, selects only the segments in which the flight level decreased.

Figure 19 – Interactive filtering of trajectory segments.



Visualization of trajectories on a map is affected by the segment filtering shown in Fig. 19.

Figure 20 – Effect of interactive filtering of trajectory segments.

Map appearance when only horizontal segments (i.e., where the flight level did not change) are filtered out while the segments going up or down are visible.

Figure 21 – Trajectories with horizontal flight segments hidden.

Other useful interactive techniques include transformation of time references, which was illustrated in Fig. 18, and modification of visualization parameters, such as colours of bars (Fig. 7) and colours representing categories (Fig. 3), minimal and maximal line widths (Figs. 4-6), maximal heights of bars (Fig. 7), etc. Other interactive techniques are described in the book by Andrienko (2006), and various transformations of movement data are described in the book by Andrienko et al. (2013).

3 Support of pairwise comparison of trajectories

3.1 Problem statement and background

The development of algorithms for prediction of trajectories requires supporting tools that allow developers to evaluate how well their algorithms do the predictions, which means that predicted trajectories need to be compared to real ones. For any predicted trajectory, the developer should be able to see how well it matches the corresponding real trajectory with regard to the followed route, temporal characteristics, and movement attributes, such as flight level and speed. In a similar manner, developers should be able to compare results of different algorithms or the same algorithm with different parameter settings.

Our idea for supporting such comparisons visually is to establish links between corresponding points of two trajectories and visualize differences between the spatial positions, times, and attributes of these points. The implementation of this idea requires an algorithm that links corresponding points from two trajectories, henceforth called *point matching algorithm*.

In a search for a suitable algorithm, we investigated the published literature on trajectory data analysis. We found that point-wise matching of trajectories has not been addressed as a separate research problem. Some kind of matching is used as a subtask in the task of constructing a representative trajectory for a cluster of similar trajectories (Andrienko et al. 2013, van Kreveld et al. 2015). The common approach is to link points based on their time references; however, this approach is applicable only to synchronous movements. It is not fully suitable for the purpose of comparing predicted trajectories to real ones, because real flights may be delayed, and thus the real trajectories may deviate in time, sometimes quite substantially, from the predicted ones.

For our purpose, the matching between points needs to be done according to the positions of the points along the flight route, irrespective of the times when the points were attained (it is supposed that the time differences are examined after the points are matched). At the first glance, it seems that the dynamic time warping algorithm (Berndt and Clifford 1994) can be adapted for this purpose. However, after a careful study, we judged this approach as unsuitable because it can match a single point of one trajectory to multiple points of another trajectory.

Since none of the pre-existing approaches can do the required work, we developed a new algorithm, which is described in the following section.

3.2 Computation of pairwise differences

To enable pairwise comparison of trajectories, we have developed a tool that does the following for each pair of trajectories that need to be compared:

1. Finds pairs of matching points along the routes followed. Two points can be matched if they have close geographic positions as well as similar relative positions within the respective trajectories. For some points, there may be no counterparts in the other trajectory.
2. For each pair of matched points, computes the spatial distance, difference in time, and differences in values of positional attributes, such as altitude or flight level. The computed distances and differences are attached to the points as new positional attributes.
3. For each pair of matched points, builds a vector (line) connecting them.

The pseudocode of the point matching algorithm is presented below. The function *next_candidate_pair* (P, Q, M, i, j) takes care of different inter-point distances in two trajectories P and Q. If P[i] and Q[j] are two points from these trajectories considered for matching, the function compares their spatial distances to the last matched points from the respective trajectories. If one of the distances much larger than the other (the distance ratio exceeds a tolerance threshold), the function takes the following points from the other trajectory until the distance ratio fits below the threshold.

The main purpose of the function *next_candidate_pair* is to increase the algorithm performance by proposing pairs that have good chances to be matched, which decreases the number of unproductive matching attempts, when initial matches are later overridden by better matches. Ideally, candidate points for matching are on the same distance from the previous matched points, but this may rarely happen; therefore, some tolerance is needed. In other words, the ratio between two distances does not need to be exactly 1 but should be close to 1. When the ratio differs much from 1, it is reasonable to check whether the next point in the trajectory where the distance is smaller can be a better candidate. We use the tolerance threshold 1.2 for the ratio of the larger distance to the smaller one. It means that the function will check the suitability of the next point only when the distance ratio exceeds 1.2. We would like to note that variation of the value of the tolerance threshold does not have an effect on the algorithm results, i.e., the set of pairs of matched points, because for each candidate pair the algorithm checks if better matches exist. The choice of the threshold value can only affect the number of initial suboptimal matches that are later overridden by better ones. Specifically, increasing the threshold increases the chances that suboptimal pairs will be proposed for matching. On the other hand, decreasing the threshold requires the function *next_candidate_pair* to do more checks. As a result, the overall impact on the performance is not large. In our experiments, we found that the threshold value 1.2 works well enough and does not require fine tuning.

The presented algorithm performs tentative matching. After that, the following post-processing is done: for unmatched points from each trajectory, it is checked if it can be a better match than one of its neighbours to the neighbour's counterpart. If so, the match is updated.

Given:

- Trajectories P and Q
- Tolerance threshold TT, to account for differing inter-point distances in two trajectories.

Output:

– List of matching point pairs M

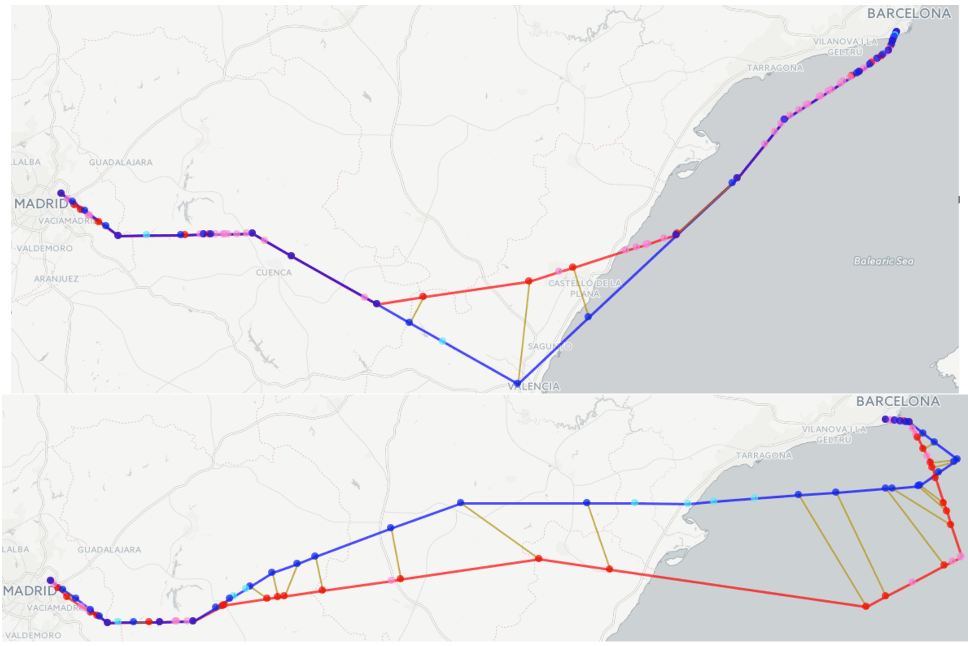
Description of the algorithm:

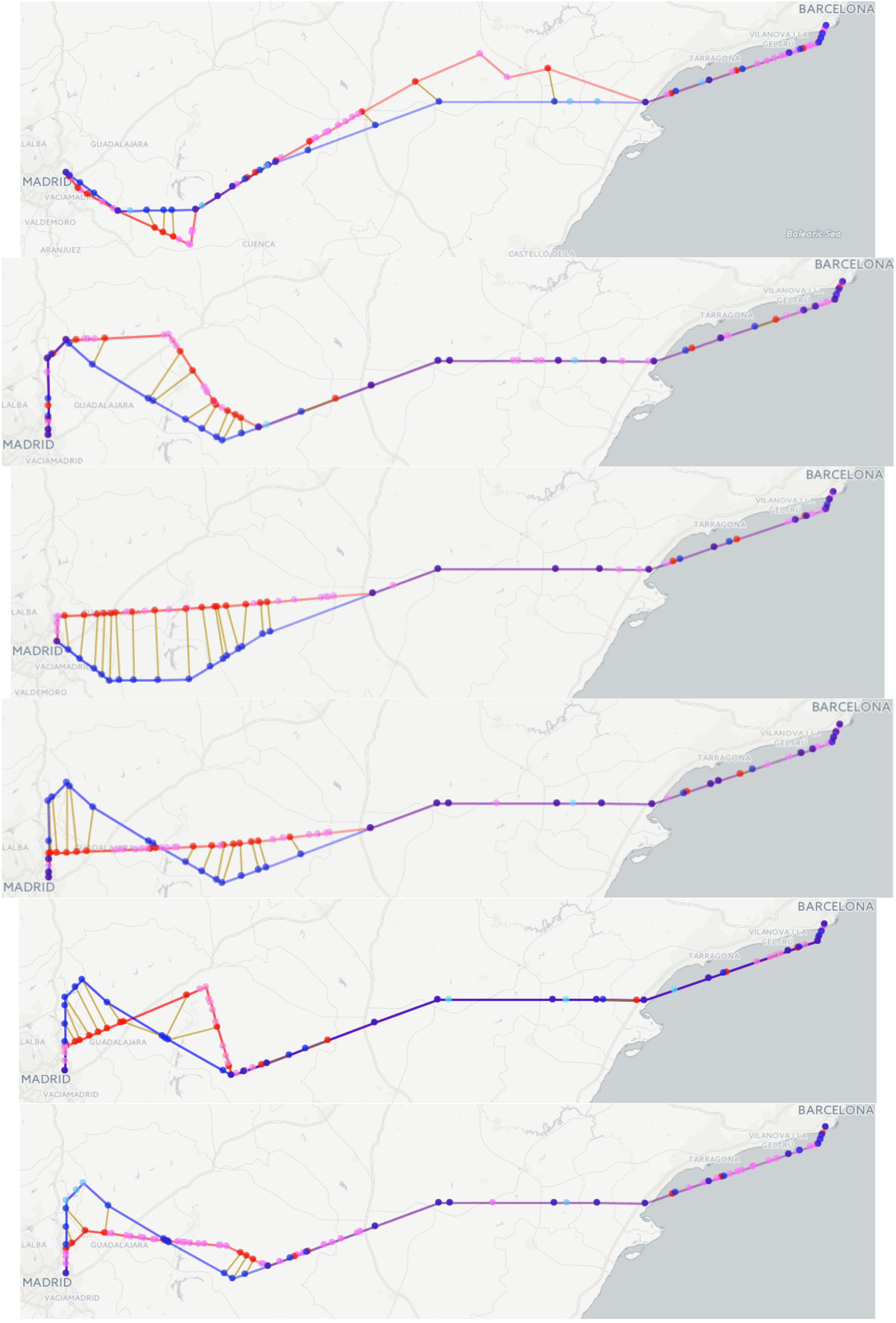
```
let M = <(1,1)>;                                    /* list M contains the pair (1,1) */
let  i = 2; let  j = 2;
while  i <= P.length and  j <= Q.length  do         /* scan the trajectories from the start to the end */
   let  cp = next_candidate_pair (P, Q, M, i, j);   /* account for different inter-point distances, if needed */
   let  i = cp.first; let j = cp.second;            /* indexes of currently considered points in P and Q */
   let mi = M.lastElement[1]; let mj = M.lastElement[2];   /* indexes of the last matched points from P and Q */
   let n = argmin(distance(P[i],Q[j]), distance(P[i], Q[mj]), distance(P[mi], Q[j]), distance(P[mi], Q[mj]));
    if n = 2 and                                    /* point P[i] is a better match to point Q[mj] than P[mi]  */
      distance(P[i], Q[mj]) < distance(P[i], Q[j+1])   /* point P[i] does not match Q[j+1] better than Q[mj]  */
    then
        let  M = M – (mi, mj) + (i, mj);            /* replace the pair (mi, mj) in M by (i, mj)  */
        let  i=i+1;                                 /* proceed to the next point in P */
    end_if;
    else
       if n = 3 and                                 /* point Q[j] is a better match to point P[mi] than Q[mj]  */
         distance(P[mi], Q[j]) < distance(P[i+1], Q[j])   /* point Q[j] does not match P[i+1] better than P[mi]  */
       then
           let  M = M – (mi, mj) + (mi, j);         /* replace the pair (mi, mj) in M by (mi, j)  */
           let j = j +1;                            /* proceed to the next point in Q */
       end_if;
       else
          let n = argmin(distance(P[i],Q[j]), distance(P[i], Q[j+1]), distance(P[i+1], Q[j]));
          if n = 2 and                              /* point Q[j+1] is a better match to point P[i] than Q[j] */
            distance(P[i], Q[j+1]) < distance(P[i+1], Q[j+1])   /* point Q[j+1] is not a better match to P[i+1] than to P[i] */
          then
              let j = j +1;                         /* proceed to the next point in Q */
          end_if;
          else
             if n = 3 and                           /* point P[i+1] is a better match to point Q[j] than P[i] */
               distance(P[i+1], Q[j]) < distance(P[i+1], Q[j+1])   /* point P[i+1] is not a better match to Q[j+1] than to Q[j] */
             then
                 let i = i +1;                      /* proceed to the next point in P */
             end_if;
             else
                 let  M = M + (i,j);                /* add the pair (i, j) to M */
                 let i = i +1;  let j = j +1;       /* proceed to the next points in P and Q */
             end_else;
          end_else;
       end_else;
    end_else;
end_while;
return M;
```

```
function next_candidate_pair (P, Q, M, i, j):
if i > P.length  or  j > Q.length  return null;
let mi = M.lastElement[1];  let mj = M.lastElement[2];           /* (mi, mj) is the last matched pair */
let d1 = distance(P[i], P[mi]);  let  d2 = distance(Q[j], Q[mj]);    /* inter-point distances in P and Q */
if  d1 > d2 then
      while d1 > d2 * TT and  j < Q.length  do                    /* much larger inter-point gap in P */
            let j = j+1;  let  d2 = distance(Q[j], Q[mj]);        /* proceed to the next point in Q */
      end_while;
end_if;
else
      while d2 > d1 * TT and  i < P.length  do                    /* much larger inter-point gap in Q */
            let i = i+1;  let  d1 = distance(P[i], P[mi]);        /* proceed to the next point in P */
      end_while;
end_else;
return (i, j);
```

The images in Fig. 22 illustrate how the algorithm matches points in pairs of trajectories. In this example, trajectories built from flight plans (blue) are compared with the trajectories of the corresponding actual flights (red). The points of the actual trajectory that have matching points in the planned trajectory are represented by dots coloured in red and their counterparts by dots coloured in blue. The matching points are connected by dark yellow lines. The points that have no matches are represented by dots coloured in pale pink colour in the actual trajectory and pale cyan colour in the planned trajectory. Please note how the algorithm copes with the large differences between inter-point distances in two trajectories.

The following Figure 22 presents examples of results of point matching between trajectories obtained from flight plans (blue) and the trajectories of the corresponding actual flights (red). The points that have been matched are represented by dots having the colours of their trajectories. The matched points are connected by dark yellow lines. The points having no matches are represented by pale coloured dots, cyan in the planned trajectory and pink in the actual one.




BARCELONA
TARRAGONA
GUADALAJARA
MADRID
VACIAMADRID
VALDEMORO
ARANJUEZ
CUENCA
Balearic Sea

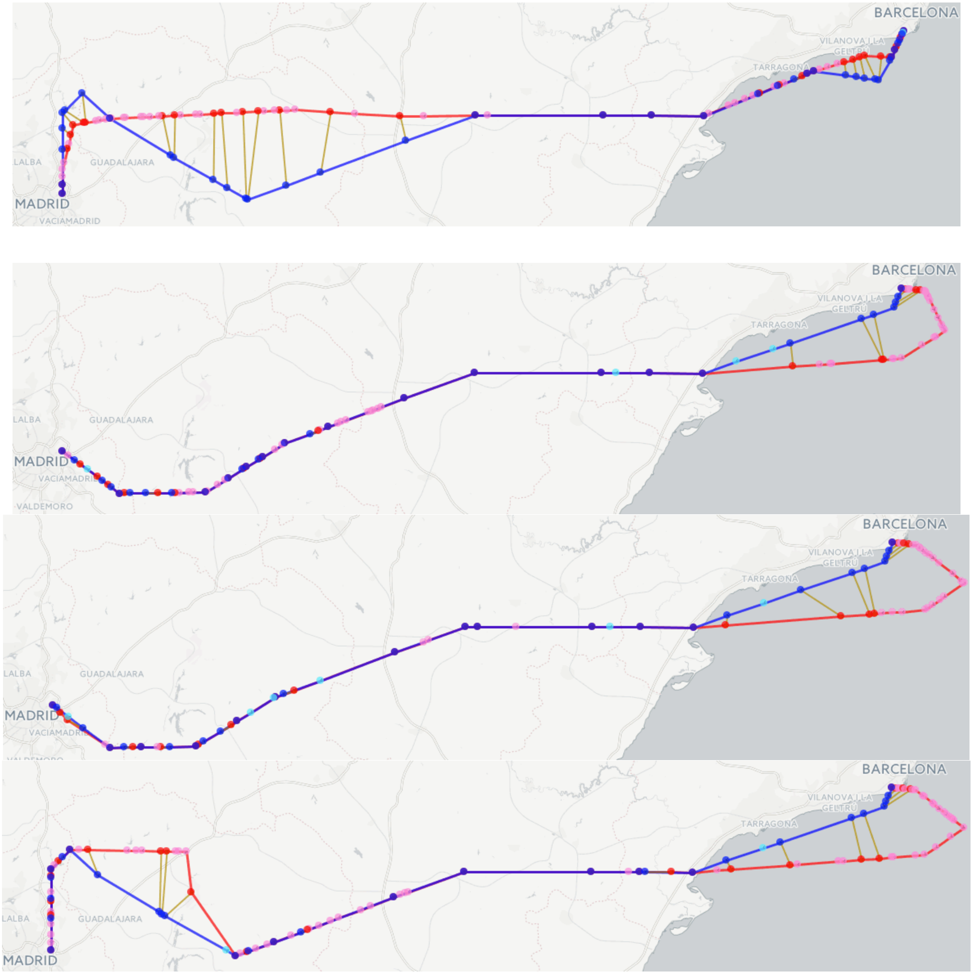


Figure 22 – Examples of results of point matching.

3.3 Interactive visually supported exploration of pairwise differences between trajectories

As is seen from Fig. 22, pairwise comparison of trajectories is facilitated by vectors that link corresponding positions. It should be taken into account that the vectors are visible only when there is a noticeable distance between the matched points. It may also be important to see which points have not been matched. Points of trajectories can be made explicit by applying some visualization to one or more attributes associated with the points. Thus, in Fig. 22, combinations of values of two attributes 'label' (with values 'actual' and 'planned') and 'has match' (with values 'true' and 'false') are represented by colour coding. In Fig. 23, values of the attribute 'match point N' are represented by shading. Value -1 signifies absence of match, and it is represented by light blue colour. The remaining values are represented by shades of brown.

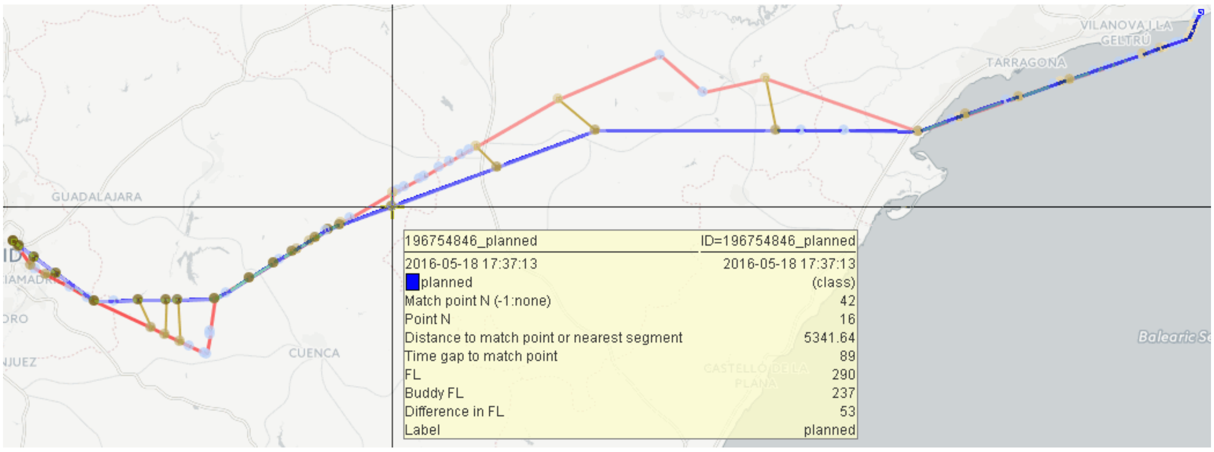
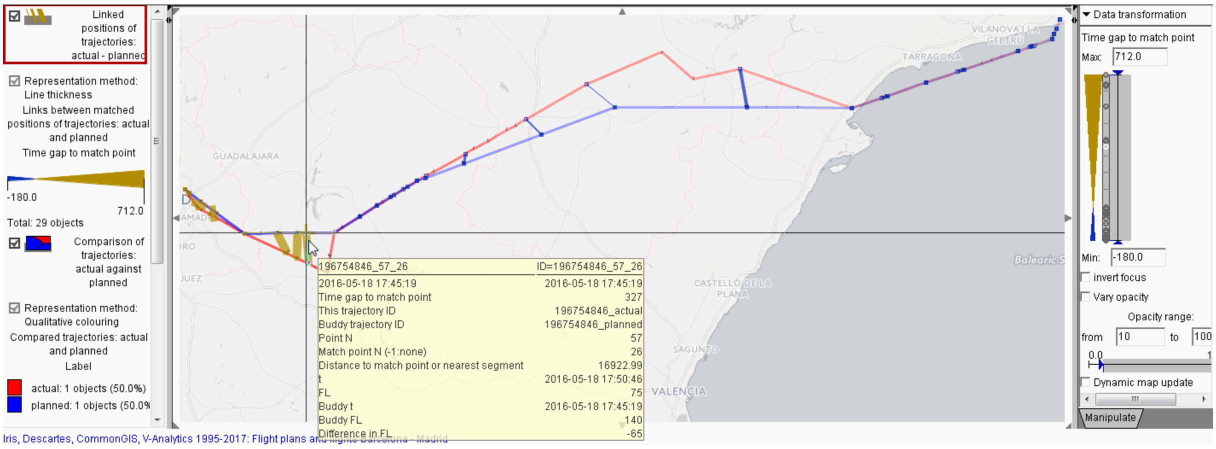


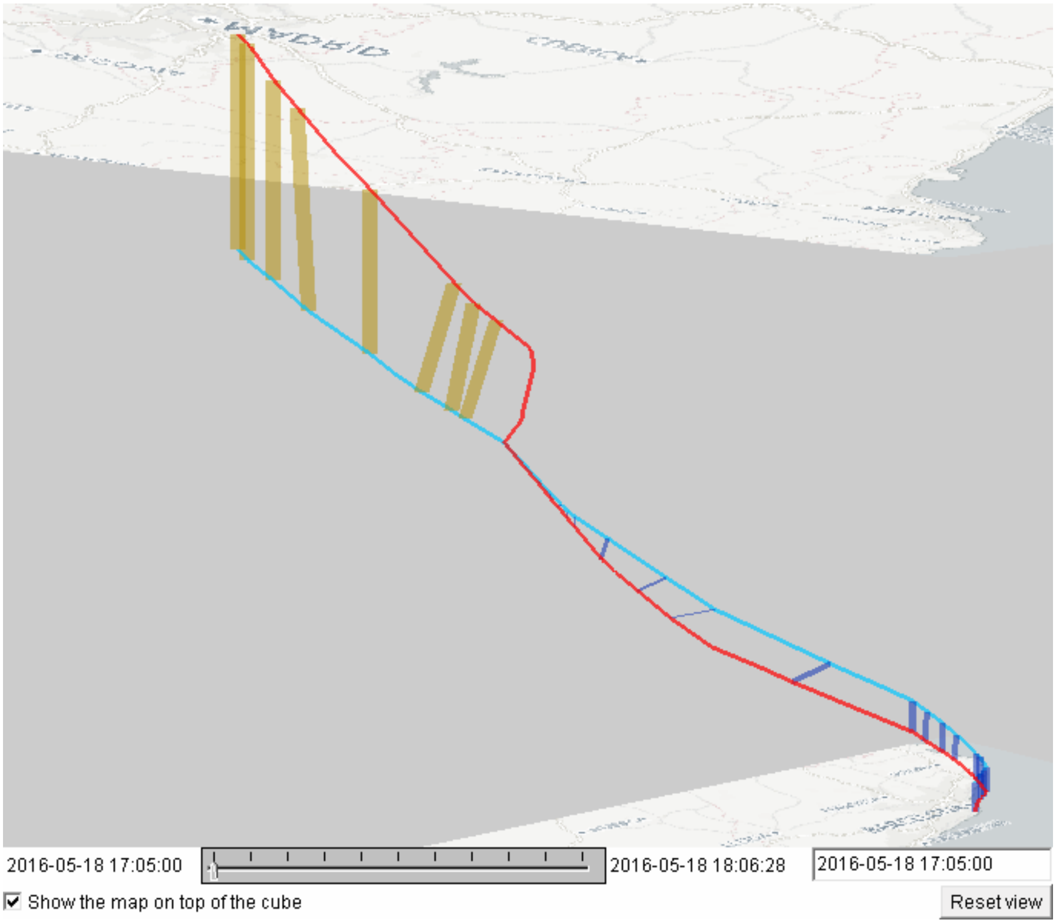
Figure 23 – Interactive access to attributes of trajectories and their points.



The widths of the link lines are proportional to the temporal distances between the matched points. Dark yellow lines represent to positive differences, which mean that the positions in the actual trajectory were attained later than the corresponding positions in the flight plan. Dark blue corresponds to negative differences.

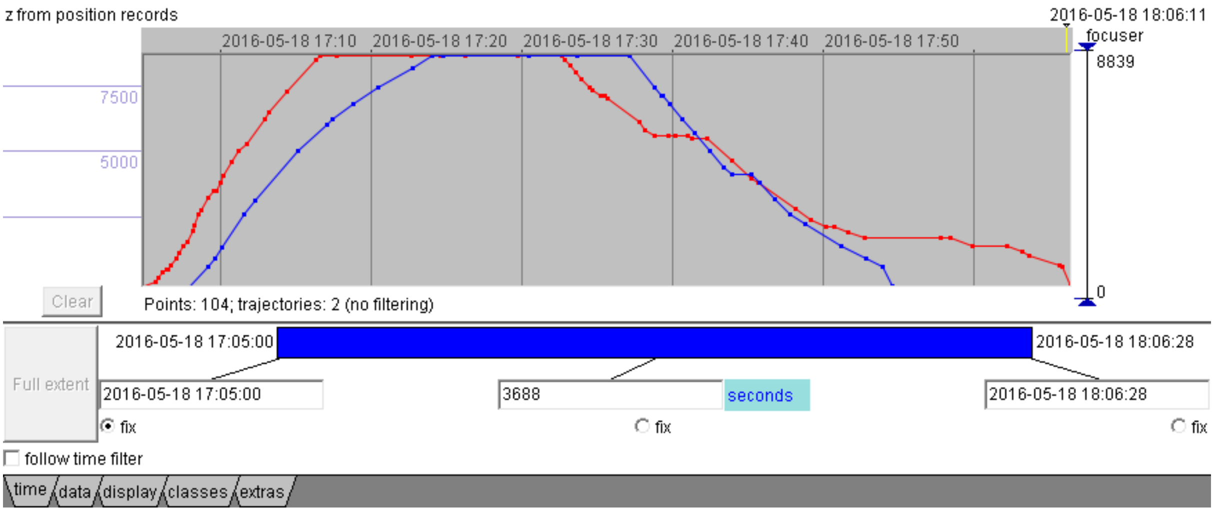
Figure 24 – Interactive access to attributes of the links between the matched points.

The values of the attributes that have been attached to the trajectory points based on the results of matching can be accessed by pointing on the points with the mouse cursor. The attribute names and values appear in a popup window, as shown in Fig. 23. The attributes include the distance in space to the nearest point or segment of the counterpart trajectory, the time gap to the matching point (if it exists), and the differences in the flight level and/or altitude (only if a match exists). The links connecting the matched points also have attributes, which can be interactively accessed (Fig. 24) and visualized. Thus, in Fig. 24, the widths of the link lines are proportional to the temporal distances between the matched points. Dark yellow lines represent to positive differences, which mean that the positions in the actual trajectory were attained later than the corresponding positions in the flight plan. Dark blue corresponds to negative differences. In the example shown in Fig. 24, the actual flight was initially ahead of the planned time for 3 minutes (180 seconds), then the difference was decreasing as the flight first deviated to the north from the planned route and then returned back. In the last 20 minutes, the flight sharply deviated to the south, and the temporal differences turned to positive, i.e., the actual flight turned to be behind the planned time. The eventual delay was about 12 minutes (712 seconds). The temporal differences can also be seen in a space-time cube (Fig. 25), which is especially useful when there are little or no differences between the followed routes.



A space-time cube shows temporal differences of an actual flight (red) from the flight plan (light blue). The connecting lines in blue correspond to the actual flight being ahead of the plan, and the lines in dark yellow to the actual flight being behind the plan. The line thickness is proportional to the absolute time difference.

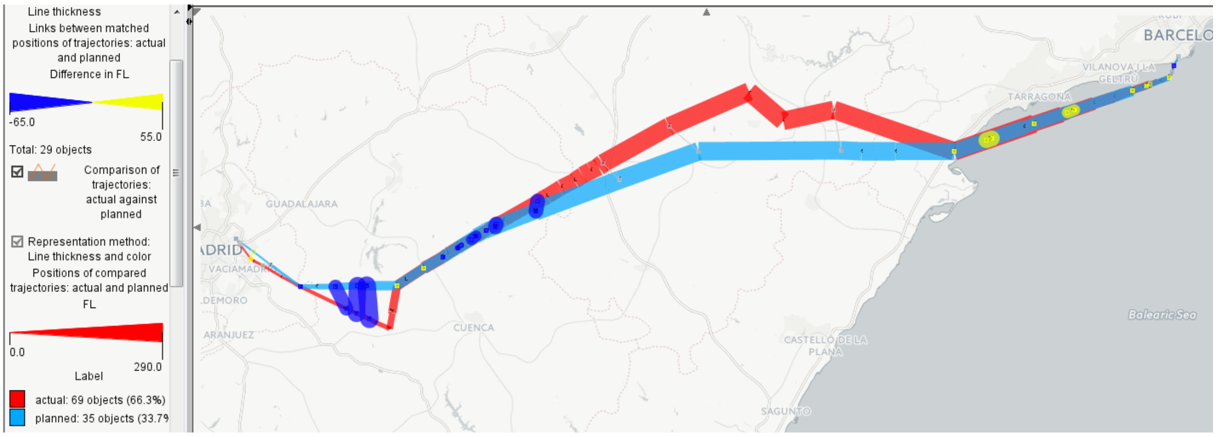
Figure 25 – Temporal differences shown in a space-time cube.



The temporal development of an actual flight (red) in comparison to the plan (blue) can be compared with the use of a time graph showing the dynamics of a chosen numeric attribute; here: altitude.

Figure 26 – Comparison of the temporal developments of planned and actual flights using a time graph.

The positional attribute values and their temporal dynamics in two trajectories can be compared using a time graph. Thus, the graph in Fig. 26 represents the dynamics of the altitude in the actual flight (red) and the plan (blue) that were used in the previous figures 23-25.



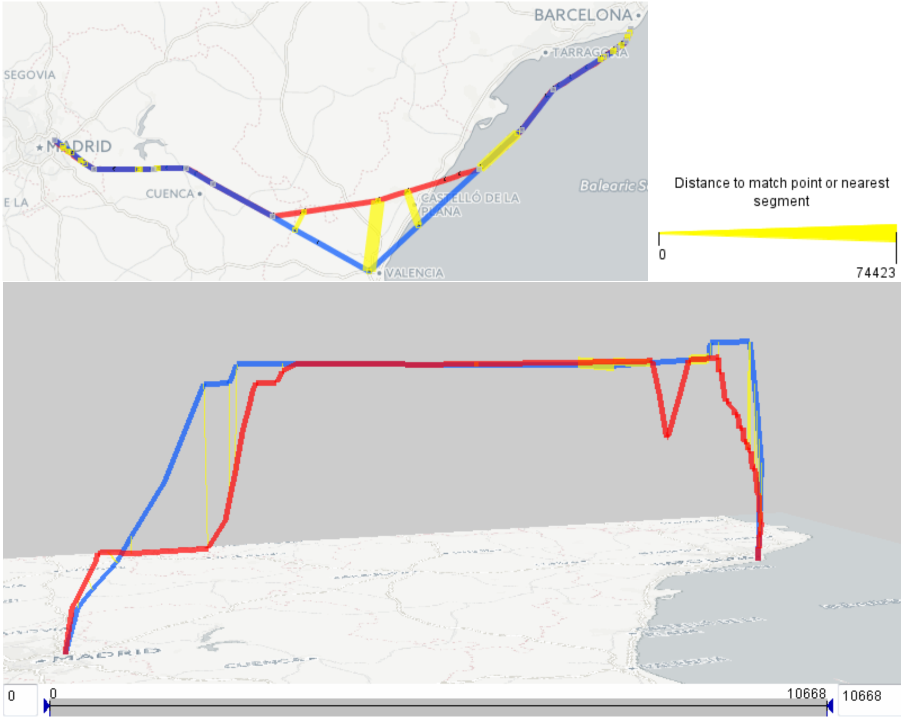
The line segment widths in the trajectories are proportional to the flight levels. The widths and colours of the linking lines between the trajectories represent the differences in the flight levels between the actual flight and the plan. Lines in yellow represent positive differences (the actual flight level was higher than planned) and lines in dark blue represent negative differences (the actual flight level was lower than planned).

Figure 27 – Comparison of planned and actual flights using a map.

Figure 27 demonstrates the possibility to visualize positional attributes of trajectories by varying the appearance of the line segments between the points. In this example, the segment widths are proportional to the flight levels attained in the starting points of the segments. The colours (red or light blue) differentiate the actual trajectory from the planned. The widths and colours of the linking vectors represent the differences between the flight levels of the matched points. Yellow corresponds to higher actual flight levels than planned, and dark blue corresponds to lower actual flight levels than planned.

The comparison of the flight levels or altitudes is also supported by a 3D display of trajectories and links between them (Fig.28). The visualization is similar to the space-time cube, but the vertical display dimension is used to represent the altitudes. Several examples are shown below. Each screenshot of the 3D display shows a pair of compared trajectories with linking lines between the matched points. The same trajectories and links are also shown on a 2D map placed above the 3D view. The widths of the linking lines in both the map and the 3D view are proportional to the horizontal geographical distances between the points.

In the following Figure 28, each pair of images shows a pair of compared trajectories with links between their matched points. The trajectories and links are represented on a map in the upper image and in a 3D view in the lower image; the vertical dimension represents the flight altitudes. Blue and red colouring is applied to planned and actual trajectories, respectively; in yellow are the links between the points. The widths of the linking lines are proportional to the horizontal geographical distances between the corresponding points.


BARCELONA
SEGOVIA
MADRID
CUENCA
VALENCIA
Balearic S
Distance to match point or nearest segment
0
74423
0
0
10668
10668


BARCELONA
SEGOVIA
MADRID
CUENCA
Balearic Sea
Distance to match point or nearest segment
0
74423
0
0
10668
10668

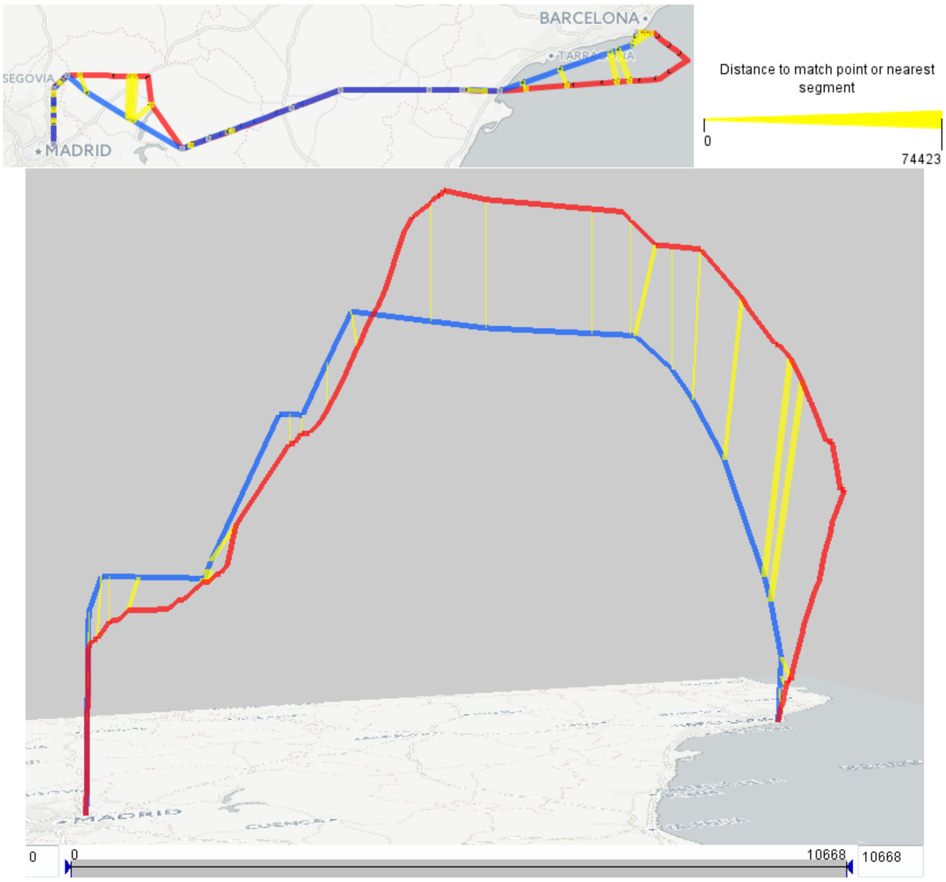


Figure 28 – Comparison of flight levels using a 3D display.

4 Support of exploration of multiple comparison results

4.1 Statistics of point-wise differences

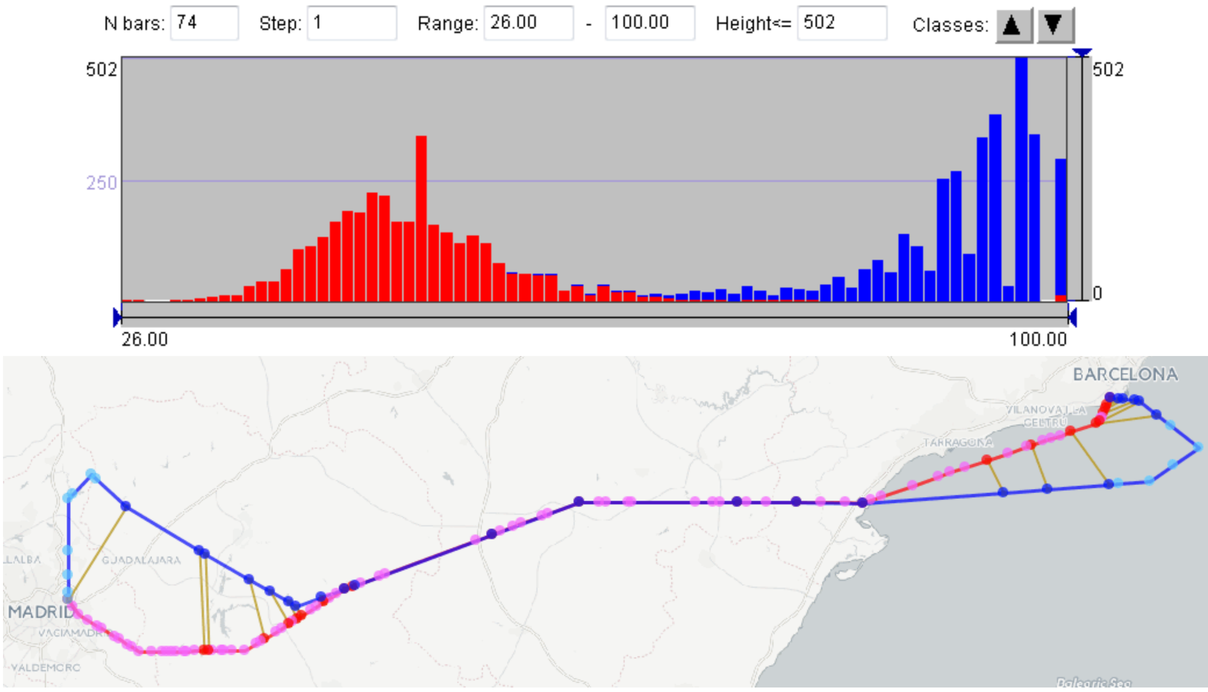
In addition to positional attributes reflecting matches and differences between points of trajectories, the matching tool computes for each trajectory summary statistics of these point-wise matches and differences. The statistics include

- the count of matched points and the proportions of this count to the lengths of both trajectories and to the sum (count_matched + count_unmatched_1 + count_unmatched_2);
- the mean and maximal distances between matched points;
- the mean and maximal distances of unmatched points to the nearest point or segment from the other trajectory;
- the mean and maximal time gaps between matched points;
- the time gaps between the start points and between the end points;
- the length of the common path (= the path built from the mean points of the matched points) and its proportion to the lengths of both trajectories.

These statistics are attached to the trajectories as values of new attributes, one attribute for each statistics. After performing pairwise matching for a set of trajectories, it is possible to visualize any of these attributes in aggregated form, such as frequency histograms, and explore the statistical distribution of the differences. Trajectories with extreme values of some of the attributes can be interactively selected and explored in detail. Below are some examples for a set of 3,410 flights from Barcelona to Madrid, where for each flight there is an actual trajectory and an expected trajectory according to the flight plan. The matching tool has been applied to each pair of actual and planned trajectory. In all histograms and maps, red and blue colours correspond to the actual and planned trajectories.

The frequency histogram in Fig. 29 shows the statistical distribution of the proportion of the matched points to the total number of points in a trajectory. It can be seen that proportions are higher for the planned trajectories (blue) than for the actual trajectories (red). The reason is that the flight plans in the studied dataset are described by a smaller number of points than the actual flights. Hence, even when an actual flight is done exactly according to the plan, there will be additional points in the actual trajectory for which there are no corresponding points in the planned trajectory.

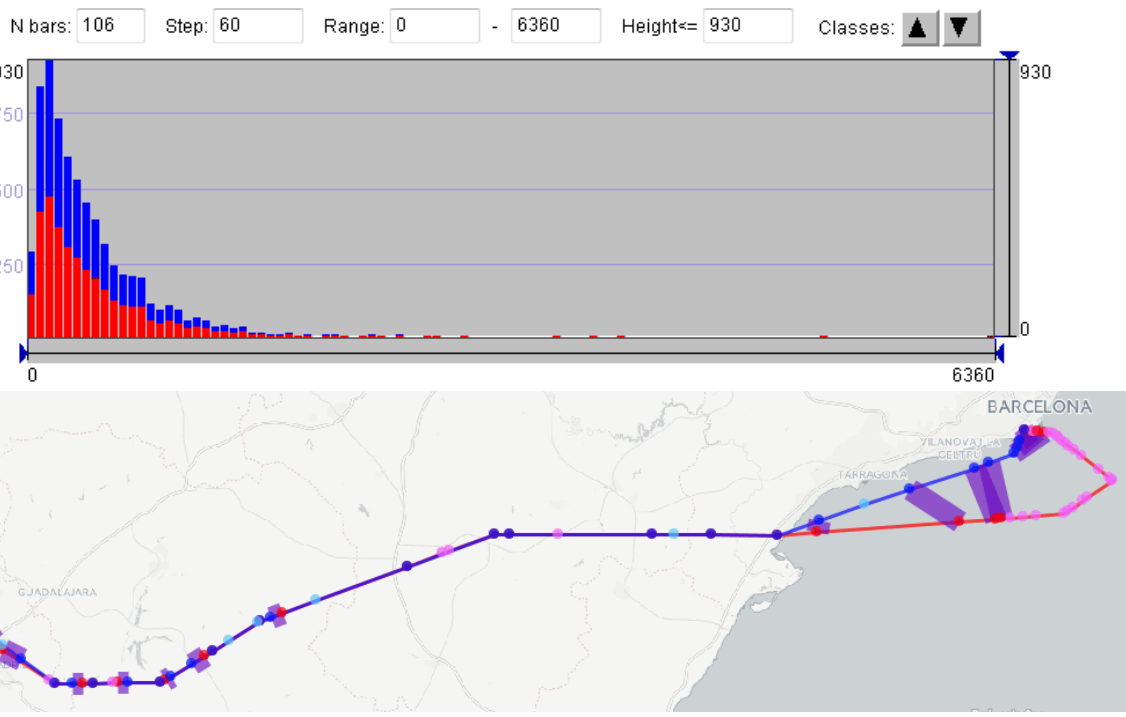
The histogram may exhibit the presence of extremely low values of the proportion of the matched points. Using interactive filtering, trajectories with these values can be selected for detailed inspection. Thus, the lower image in Fig. 29 shows a pair of planned vs. actual trajectories where only 26% of points in the actual trajectory were matched to points of the planned trajectory. It can be seen that the actual trajectory has much higher frequency of points than the planned trajectory. Besides, the actual flight substantially deviated from the plan; therefore, some points in the planned trajectory have no matches in the actual trajectory.



The histogram shows the statistical distribution of the proportions of the matched points. The map shows the pair where the actual trajectory has the smallest proportion of the matched points (26%).

Figure 29 – Exploration of the proportions of matched points.

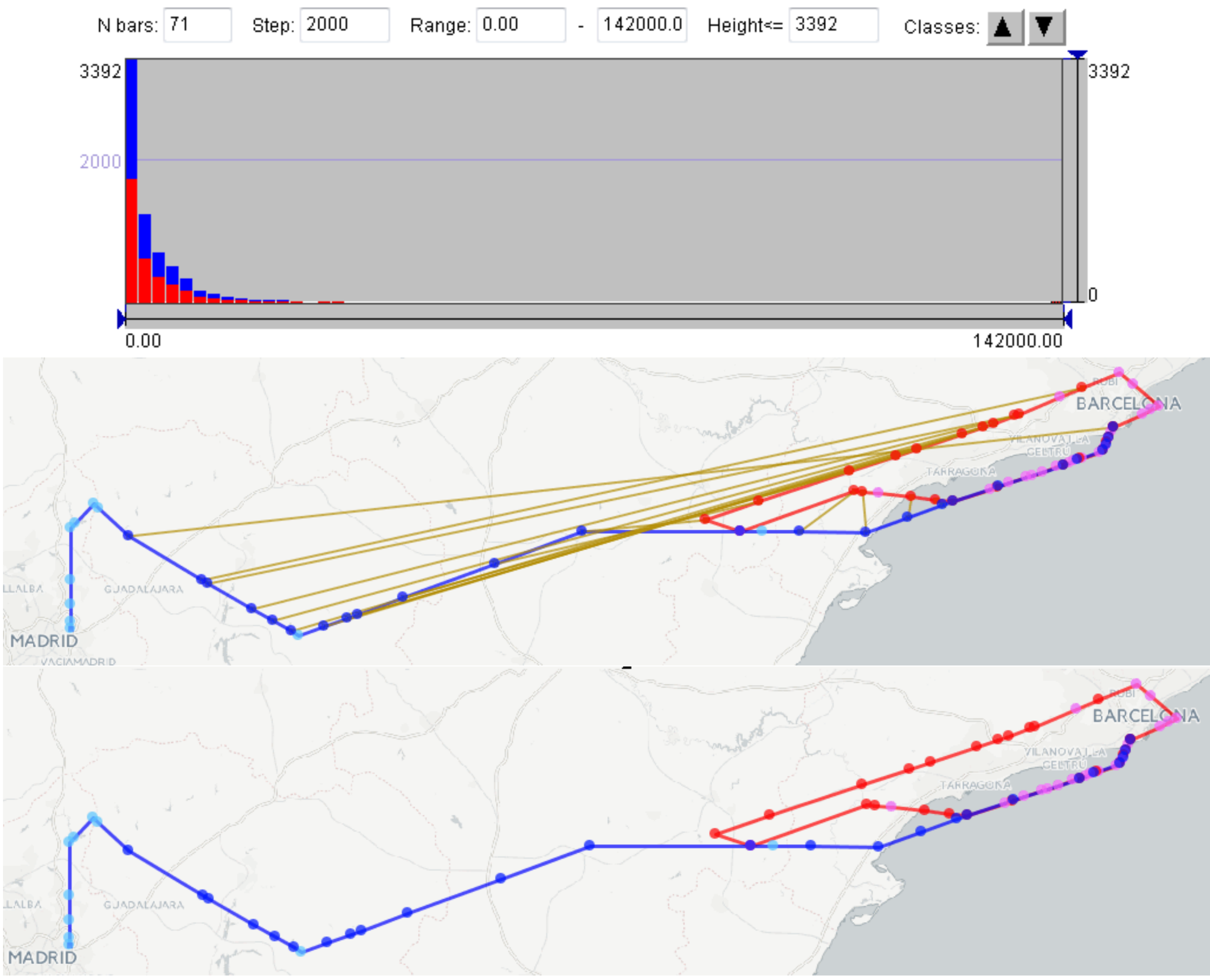
The histogram in Fig. 30 shows the statistical distribution of the mean time differences between the matched points from the actual and planned trajectories. While the most frequent values are about 1-3 minutes, there are few trajectories with extremely high values, the maximum reaching 6,360 seconds (106 minutes). The map in Fig. 30 shows the pair of trajectories for which this maximal mean time gap was attained. It was mostly caused by a delayed departure of the actual flight. At the beginning, the actual flight deviated from the plan, which, apparently, increased the initial delay.



The histogram shows the statistical distribution of the mean temporal gaps between the matched points. There are a few outliers with extremely high values. The map shows the pair of trajectories for which the highest value was attained. The widths of the linking vectors between the positions are proportional to the time gaps between the linked points. In this example, the actual flight started later than planned by 96.27 minutes. The largest point-wise time gap was 111.7 minutes. By the end of the flight, the delay was 110.8 minutes.

Figure 30 – Exploration of the temporal gaps between matched points.

Figure 31 demonstrates another extreme case where there is very large spatial deviation of the actual flight from the planned. The histogram shows an outlier among the mean distances between the actual and planned trajectories that greatly differs from the remaining values. The corresponding pair of trajectories is shown on a map. It can be seen that in the actual flight the plane returned back to the departure airport.



The histogram shows the statistical distribution of the mean distances between the matched points. There is an outlier with an extremely large mean distance (about 142 km). The map shows the pair for which this mean distance was attained. The lower image shows the trajectories without the linking vectors between the points. It is clearly seen that the actual flight (red) returned back to Barcelona.

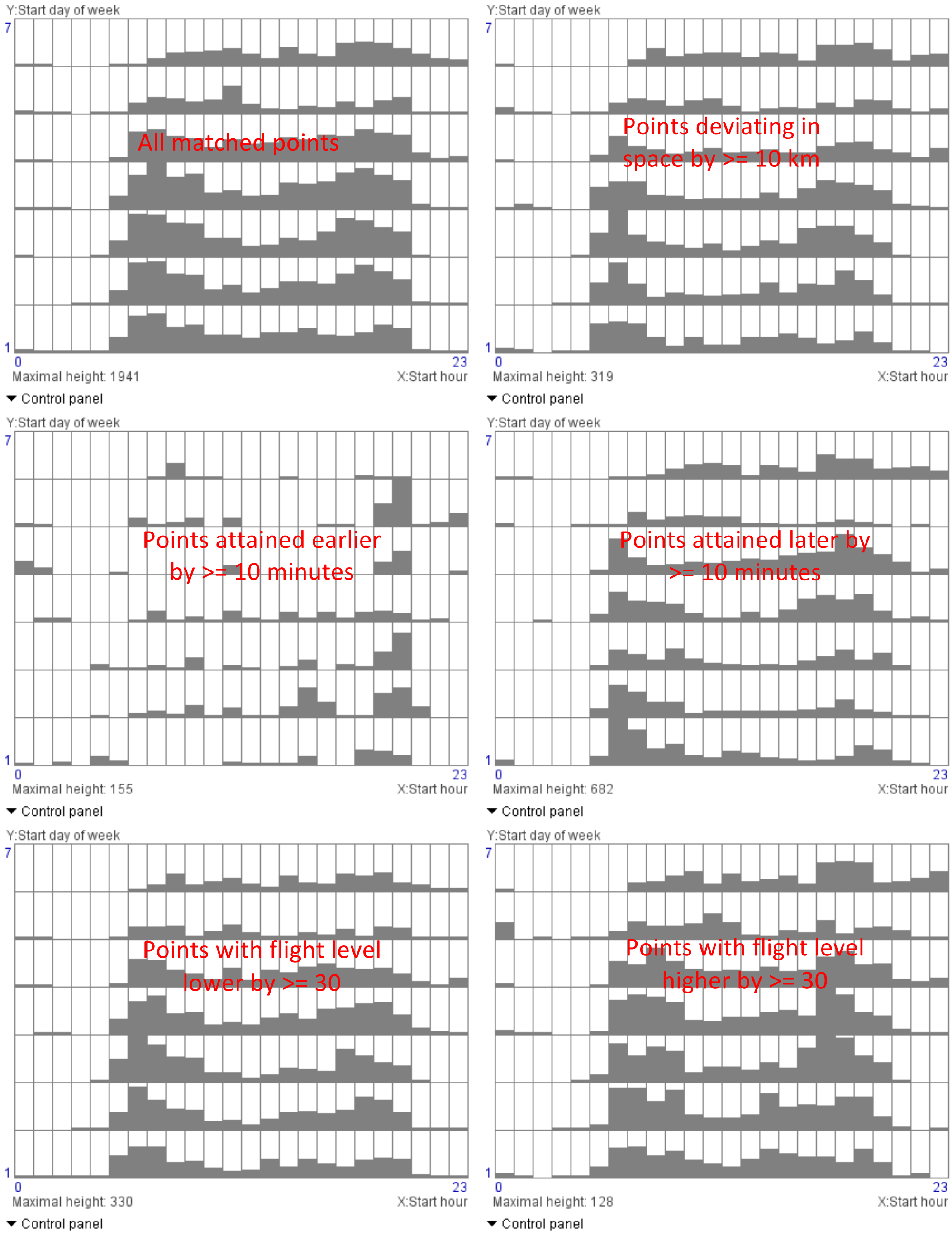
Figure 31 – Exploration of the spatial distances between matched points.

4.2 Exploration of the spatial and temporal distribution of deviations

The previous section demonstrated the possibilities for detection and detailed exploration of very large differences between two variants of trajectories: actual vs. planned, actual vs. predicted, predictions obtained from different methods or with different parameter settings, etc. Besides this kind of analysis, it may also be reasonable to investigate the temporal and spatial distributions of the differences. Observed patterns may, in particular, show where and when the predictions are not good enough and need improvement, or where and when the actual flights tend to deviate from the plans, which may indicate the directions for improving the flight planning.

Figures 32 and 33 show examples of the temporal and spatial distributions of actual flights' deviation from their corresponding flight plans, for a set of 3,410 flights from Barcelona to Madrid. The exploration is applied to the matched points of the actual trajectories. The 2D time histogram in Fig. 32 shows the distribution of these points over the days of the week (vertical dimension; value 1 at the bottom corresponds to Monday and value 7 at the top to Sunday) and hours of the day (horizontal dimension; value 0 corresponds to midnight). The image on the top left represents the distribution of the whole set of matched points. Using interactive filtering, it is possible to select various subsets of points. In response to each filtering operation, the histogram changes to show the distribution of the chosen subset. The remaining images in Fig. 32 correspond to different selections of subsets of points. Thus, the top right image shows the temporal distribution of the points that deviate in space by 10 km or more from the corresponding planned positions. Many such deviations occurred in hour 6 (from 06:00 to 07:00 on Tuesday and Wednesday. The image in the middle left represents the points with negative temporal deviations from the planned positions, i.e., the positions were attained earlier than planned. The majority of these cases occurred in hour 20 (from 20:00 to 21:00), especially on Saturday. The image in the middle right represents the positions that were later than planned. These positions are more numerous; the peaks occur in hour 6 in the week days and in hour 18 on Friday. The images at the bottom represent the distributions of the negative and positive deviations from the planned flight levels.

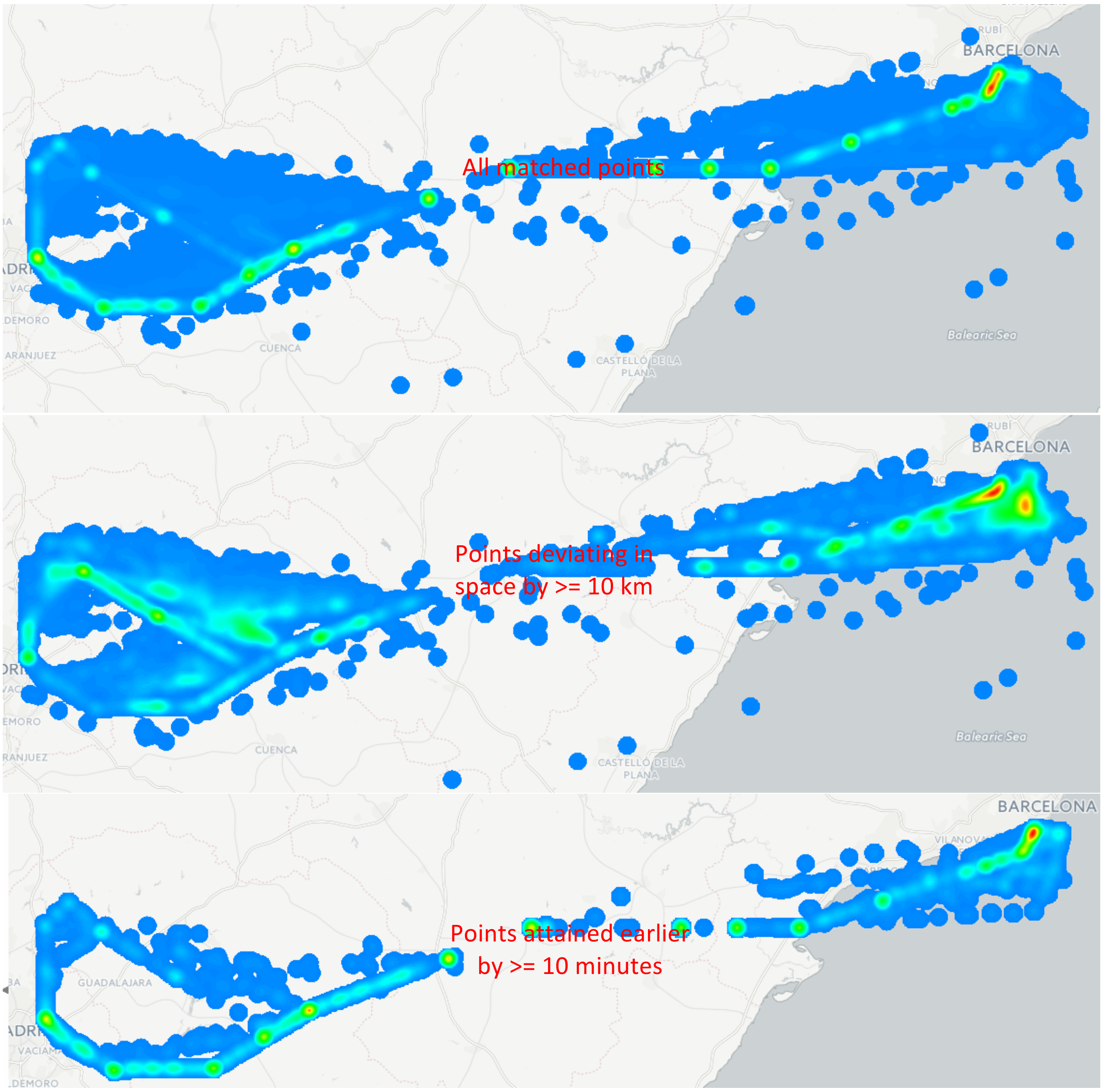
The images in Fig. 33 show the spatial distributions of the same subsets of points as in Fig. 32. The spatial distributions are represented by density maps, where the colour scale from blue to red represent the densities from low to high. The first image shows the density distribution of the whole set of matched points from the actual trajectories. The second image shows where the points with high spatial deviations from the planned routes mostly occur: on the departures from Barcelona towards the south and on the approaches to Madrid before landing from the north in the southern direction. The next two images correspond to the points with the negative and positive temporal deviations with respect to the plans. The spatial patterns are very similar to the pattern for the whole set of points; hence, the temporal deviations are not related to the followed routes. The last two images represent the spatial distributions of the negative and positive deviations from the planned flight levels. It is seen that negative deviations (flying lower than planned) occurred more frequently in the last thirds of the flights approaching Madrid from the south while positive deviations occurred more frequently in the first thirds of the flights, especially at the position of crossing the shoreline.



Two-dimensional histograms show the temporal distribution of the matched points from the actual trajectories by the hours of the day (horizontal dimension) and days of the week (vertical dimension). Top left: the distribution of all matched points. Top right: the distribution of the points that deviate in space from the corresponding planned positions by 10 km or more. Middle left: points that were earlier than planned by 10 minutes or more. Middle right: points that were later than planned by 10 minutes or more. Bottom left: points where the flight level was lower than planned by 30 or more. Bottom right: points where the flight level was higher than planned by 30 or more.

Figure 32 – Exploration of temporal distribution of matched points.

In the following Figure 33, density maps show the spatial distribution of the points from the actual trajectories that have matches in the planned trajectories. From top to bottom: the spatial distribution of all matched points; the distribution of the points that deviate in space from the corresponding planned positions by 10 km or more; the points that were earlier than planned by 10 minutes or more; the points that were later than planned by 10 minutes or more; the points where the flight level was lower than planned by 30 or more; the points where the flight level was higher than planned by 30 or more.



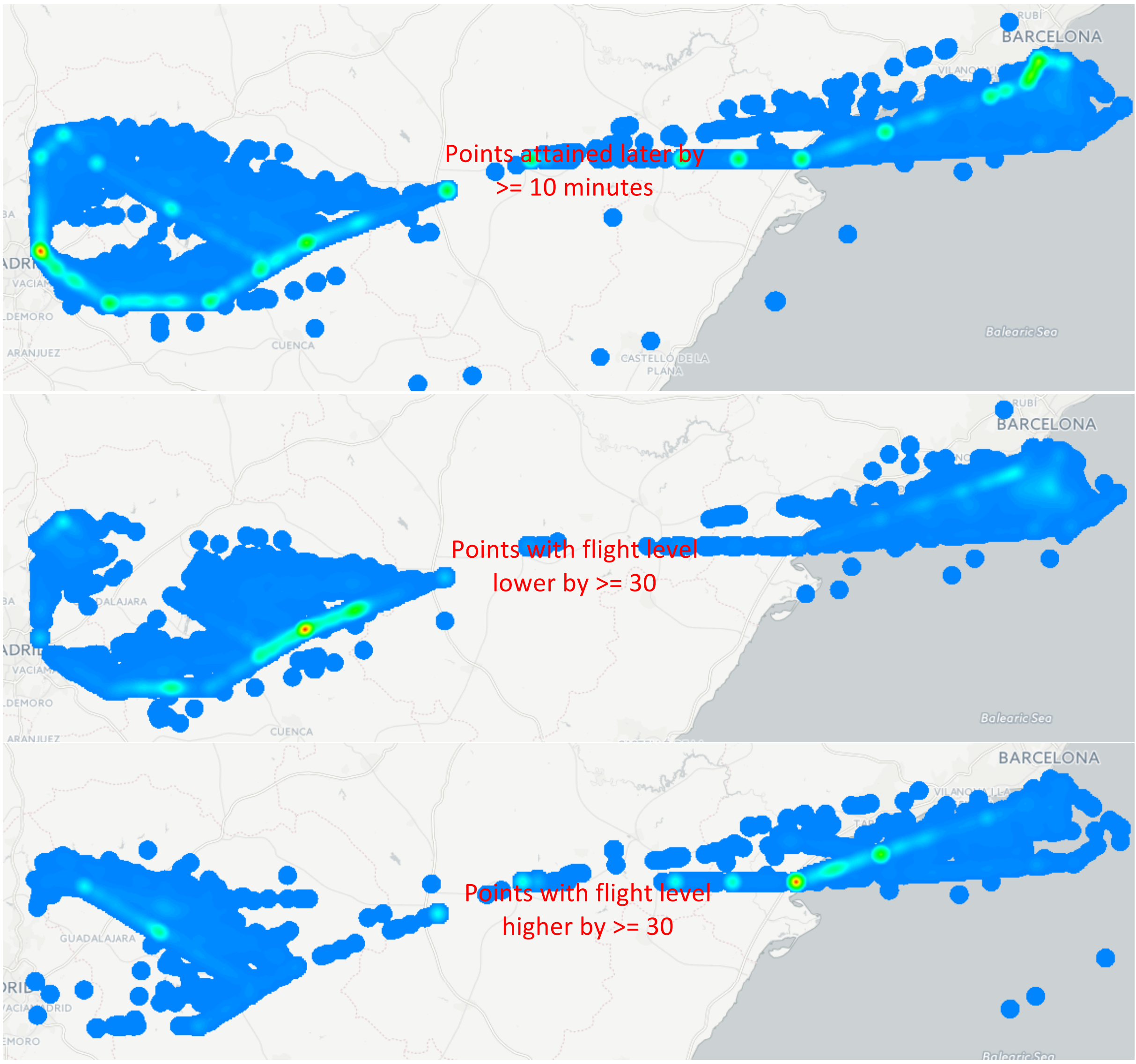


Figure 33 – Exploration of the spatial distribution of matched points.

4.3 Conclusion

For developing and evaluating algorithms for prediction of trajectories, it is important to have the possibility of detailed comparison of predicted trajectories to actual ones, to see how accurate the prediction is. It is also necessary to compare predictions obtained with different parameter settings, to understand the impact of the parameters and to choose the most suitable settings. This document presents techniques that can support these and other comparisons between trajectories, including also comparisons between flight plans and actual flights. The central technique is the point matching method, and the other techniques are interactive visual interfaces enabling the analyst to view and explore the results of point matching. We have demonstrated by numerous examples how pairwise differences can be examined in detail by looking at pairs of corresponding trajectories and how the overall temporal and spatial distributions of the detected deviations can be investigated by looking at temporal and spatial aggregations. The presented tools are capable to provide the necessary support to development of trajectory prediction algorithms.

5 References

G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel. **Visual Analytics of Movement**. Springer, 2013.

N. Andrienko and G. Andrienko. **Exploratory Analysis of Spatial and Temporal Data. A Systematic Approach**. Springer, 2006.

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.

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.

M. J. van Kreveld, M. Löffler, and F. Staals. Central trajectories. In: **31st European Workshop on Computational Geometry (EuroCG)**, Book of Abstracts, pp. 129-132, 2015