# Visual Analytics Tools for Analysis of Movement Data

Gennady Andrienko[1] Natalia Andrienko[1] Stefan Wrobel[1,2]

[1]Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS
Schloss Birlinghoven
53754 Sankt Augustin, Germany

[2]University of Bonn
Dept of Computer Science
Römerstraße 164
53117 Bonn, Germany

gennady.andrienko@iais.fraunhofer.de natalia.andrienko@iais.fraunhofer.de stefan.wrobel@iais.fraunhofer.de

## ABSTRACT

With widespread availability of low cost GPS devices, it is becoming possible to record data about the movement of people and objects at a large scale. While these data hide important knowledge for the optimization of location and mobility oriented infrastructures and services, by themselves they lack the necessary semantic embedding which would make fully automatic algorithmic analysis possible. At the same time, making the semantic link is easy for humans who however cannot deal well with massive amounts of data. In this paper, we argue that by using the right visual analytics tools for the analysis of massive collections of movement data, it is possible to effectively support human analysts in understanding movement behaviors and mobility patterns. We suggest a framework for analysis combining interactive visual displays, which are essential for supporting human perception, cognition, and reasoning, with database operations and computational methods, which are necessary for handling large amounts of data. We demonstrate the synergistic use of these techniques in case studies of two real datasets.

## Keywords

Movement data, trajectory, movement patterns, movement behavior, visual analytics, exploratory data analysis, visualization, interactive displays, cluster analysis, aggregation.

## 1. INTRODUCTION

In many areas of people's life and activities it is important to understand movement behaviors of people, animals, vehicles, particles, or other objects. Thanks to the recent advent of inexpensive positioning technologies, data about movement of various mobile objects or agents are collected in rapidly growing amounts. There is a pressing need in adequate methods for analyzing these data and extracting relevant information. The existing methods are scarce and mostly not scalable to large data volumes. An ongoing EU-funded project GeoPKDD, where we participate, aims at developing methods and tools for analysis of massive collections of movement data (http://www.geopkdd.eu).

Generally, data analysis is done for two major purposes, *understanding* and *prediction*, which are relatively independent. Thus, a predictive model (e.g. probabilistic or based on a neural network) does not necessarily need to be understood by a human. The focus of our group is developing methods to support understanding of movement behaviors and mobility patterns. Visual representations as an effective way to provide material for human's perception and reasoning play here a crucial role. However, when it is necessary to make sense of very large and/or complex data, purely visual methods are insufficient. Therefore, we suggest an analysis framework that combines interactive visual displays, database processing, and computational techniques for data transformation and analysis. Essential for analysis is the interplay and synergy of the different types of techniques.

The tools we develop are not meant for a specific application but for a wide range of analysis tasks and for as diverse types of data as movements of people, vehicles, or animals through the geographical space and movements of eyes through an image or scene in studies of human perception. This is different from such works as [13][15] where machine learning methods are devised for specific tasks such as recognition of person's significant places, activities, and transportation routines. Before presenting our work, we make a brief overview of the relevant literature.

## 2. RELATED WORK

As we have noted, visualization is essential for gaining an understanding of data and underlying phenomena. The most traditional method for the visualization of movements is arrows or flow lines drawn on a map or image [20]. In the research area of time geography [8], the technique of space-time cube was invented. Two dimensions of the cube represent geographical space and the third dimension (usually vertical) represents time. Movement behavior of an entity is shown as a three-dimensional line connecting successive positions. The inclination of a line segment indicates the speed of movement: gradual rise means high speed, steep pitch signifies slow movement, and vertical segments correspond to time intervals of no movement.

Currently, animated maps [1][2] and interactive cubes [10][11] are widely used to visualise movement data. Map and cube displays are complemented with graphs and diagrams exhibiting various aspects of the movement [6][9][11][14]. However, purely visual methods fail when it is necessary to examine movements of multiple entities and/or very long movement histories.

Most approaches to handling large amounts of movement data involve data aggregation. D.Mountain and co-authors (e.g. [6][14]) suggest several aggregation-based techniques such as temporal histogram, traffic density surface, and accessibility surface, which represents travel times from a selected location. Similar approaches are described in [7]. Unfortunately, after summarizing movement data into surfaces, one can no longer see the changes of spatial positions of entities, i.e. the very essence of movement is lost. Another approach is suggested in [5]: a convex hull containing all trajectories is built, then the central tendency and dispersion of the paths are computed, and the averaged path is represented on a map. However, this approach works well only when the entities move synchronously and follow similar routes.

Tobler [17][18] suggests that numbers of entities or volumes of materials that moved from one place to another can be visualised by means of either discrete or continuous flow maps. A discrete

map represents the movements by bands or arrows whose width is proportional to the volume moved. Continuous flow maps represent the movements as vector fields or streamlines and, unlike discrete maps, are not limited with regard to the number of different locations present in the original data. Flow maps do not reflect the temporal dimension of movement data but show cumulative movements that occurred during a certain time period. However, the concept can be extended to animated flow maps or to series of flow maps showing how the flows change over time.

Another approach to handling large amounts of data is based on filtering: a data subset is selected according to a user-specified query and then visually examined [10][21]. However, this approach does not support an overall view of the data and is therefore not sufficient for a comprehensive data exploration.

Recently, movement data are coming into the focus of research in data mining. An example is the work [12] where the approach is based on transforming movement data into sequences of symbols, which are then used as an input for the computational analysis. It is commonly recognized that proper visualization of data mining outcomes is essential for a human analyst to be able to interpret them. Moreover, interactive visual interfaces can allow the user to actively *guide* the work of computational methods, as a compensation for the computer's inability to incorporate human's tacit knowledge and relevance criteria.

We apply a multidisciplinary approach to develop a framework for the analysis of massive movement data taking advantage of a synergy of computational, database, and visual techniques [3]. We introduce our framework and demonstrate its effectiveness by examples.

# 3. EXAMPLE DATASETS

One of the example datasets we shall use consists of positions of a private car, which has been GPS-tracked from December 3, 2006 till now (the car belongs to a researcher involved in the project GeoPKDD, who voluntary collects the data and provides them to the partners for testing their methods and approaches). Currently, there are more than 100,000 positions, which are stored in a relational database. Each record includes the date and time, the latitude, longitude, and altitude of the position, the direction and speed of the movement, and a few additional fields with characteristics of the satellite signals. The temporal spacing of the records is irregular, 3 seconds on the average. The data have been recorded only while the car moved.

The second dataset consists of positions of 50 trucks transporting concrete in the area of Athens, which were GPS-tracked during 41 days in August and September 2002. There are 112,300 position records consisting of the truck identifiers, dates and times, and geographical coordinates. The temporal spacing is regular and equals 30 seconds. Unlike the previous case, the positions were recorded also when the trucks did not actually move. The data are publicly available at the URL www.rtreeportal.org.

Note that neither dataset contains explicit trips with specified origins and destinations. There are also no semantically defined places but only geographic coordinates. Hence, trips and places have to be extracted from the data by means of analysis.

# 4. MOVEMENT ANALYSIS FRAMEWORK

The framework is introduced through the consideration of generic tasks that may arise in the analysis of movement data.

## 4.1 Data Preprocessing

To facilitate analysis of movement data, we perform initial preprocessing in the database, which enriches the data with additional fields: the time of the next position in the sequence, the time interval and the distance in space to the next position, speed, direction, acceleration (change of the speed), and turn (change of the direction). This operation may take several minutes, but it is done only once. Then, the data can be filtered to remove sequences of records corresponding to absence of movement, i.e. where the distance in space to the next recorded position is below a threshold, which should be selected individually for each dataset taking into account the nature of the movement. For the first of our example datasets, such filtering is not needed since the data have been recorded only during car movement. The second dataset has been filtered using the threshold 20 meters, which reduced the size of the data to about 94,000 position records.

## 4.2 Extraction of Significant Places

One of generic tasks in analyzing movement data is to detect and interpret significant places. For the first example dataset, significant places include person's home, work, and regularly visited places such as shops. For the second dataset, significant places are depots from which the load is taken and places to which it is delivered. The knowledge of significant places can help in extraction and analysis of trips.

In many cases, the time spent in a place indicates the significance of this place. Thus, a person spends much time at home and at work. Considerable times are spent in shops, sport facilities, at doctors, etc. This indicator is not perfect: taking children from a school may not require much time but staying in a traffic jam may be quite long. Still, it is useful to consider the places of stops, especially the places of repeated stops. To interpret these places and recognize whether they are significant, the analyst may overlay them on a map or image of the area where the movement took place and look what objects are situated nearby.

In some applications, the places where moving entities are likely to stop, so-called 'points of interest' (POI), are known in advance. A database of POI may be used for automatic detection of significant places. However, such a database may be unavailable, or the concept of POI may be inapplicable (as in studies of eye movements or behaviors of animals). Besides, places significant for an individual may differ from public POI. Hence, visual inspection of places of (repeated) stops is often indispensable.

Our tools provide a convenient user interface for the extraction of the positions of stops from the database. In a preprocessed dataset, the time interval to the next position, which is available in one of the derived fields, indicates the time spent at each position. Hence, a database query may extract the records where the time interval to the next position exceeds a given threshold. The query is built and fulfilled automatically; the user only needs to specify the desired threshold. The extraction of stop positions from the database takes a very short time. The results are immediately shown on a map display as point symbols (small circles) drawn in corresponding positions (in eye movement studies, the map display may show the observed image rather than represent a geographical area). The user can do the extraction several times with different threshold values and compare the results.

After extracting the positions of stops, the analyst may wish to differentiate repeated stops from occasional ones. The map

display alone is not well suited for this: when positions of two or more stops coincide, the point symbols are drawn one on top of another. A solution may be the use of spatial clustering, which groups close positions into clusters and treats isolated positions as "noise". The clusters are then shown on the map display, where they indicate potentially significant places of repeated stops.

Out toolkit includes a clustering tool implementing the algorithm OPTICS [4]. The user needs to specify the maximum allowed distance between neighboring objects in a cluster and the minimum number of objects in a cluster. The choice of parameter values is application-specific. The analyst can try various settings since the clustering tool is easy to use and the results are immediately seen on the map display. However, the background map or image may lack relevant information or the level of detail may be inappropriate for interpreting the places of the stops and finding significant places. This is the case for both example datasets. To compensate for such deficiencies, our toolkit allows the analyst to export selected points or clusters of points to Google Earth or Google Maps, where they are overlaid on a detailed aerial image (this works only for objects having geographical coordinates). Such images are helpful when the stops occur near recognizable objects. Thus, in analyzing the first dataset, we could easily recognize two shopping centers frequently visited by the car owner. In analyzing the second dataset, we could find two places looking like truck enterprises; however, many stops have no clear landmarks nearby.

Since significant places are not always recognizable on a map or aerial image or correspond to public POI, there may be a need in additional analysis taking into account not only the geographical information. Consideration of the temporal distribution of the stops may be very helpful, especially in applications where the behaviors of moving entities are linked to temporal cycles. Thus, the movement of people is linked to the daily and weekly cycles and the movement of animals may be linked to daily and seasonal cycles. Let us demonstrate with the example of the car data how this can help in interpreting stops and finding significant places.

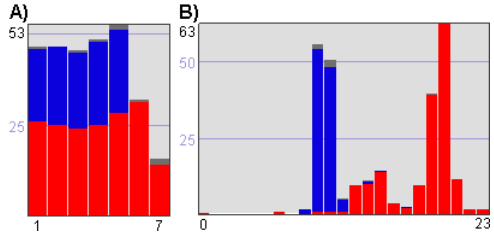


**Figure 1. The temporal histograms show the weekly (A) and daily (B) distributions of the stops of the personal car with the duration of 3 hours or more.**

We have extracted the positions of the stops lasting 3 hours or more and applied the clustering tool, which has produced two clusters with sizes 173 and 109 and classified remaining 8 stops as "noise". The results have been visualized on the map display by coloring the point symbols denoting the stop positions: each cluster is assigned a particular color, and the "noise" is shown in gray (the colors are selected automatically, but the user can easily change them). Then, we have built two temporal histograms (Fig.1) of the distribution of the stop times over the days of a week (A) and the hours of a day (B). The colors of the points on the map have been transmitted to the histograms, where the bars have been divided into colored segments proportionally to the number of the members of each cluster fitting in the respective time intervals. In Fig.1A we can see that the stops of cluster 1 (red) occur on all days of the week and the stops of cluster 2 (blue) from day 1 to day 5, i.e. from Monday to Friday. Fig.1B shows us that the stops of cluster 1 occur mostly in the second half of the day; the maximum occurrences are from 19 to 20 o'clock. The stops of cluster 2 occur mostly in the morning hours. Such a distribution makes us quite confident that cluster 1 is located near person's home and cluster 2 near person's work.

Temporal cycles often interact. Thus, in the behavior of a person, daily patterns on working days may differ from those on weekends. Our toolkit includes a tool for interactive filtering which, in particular, allows the analyst to consider separately the daily distribution of the stops occurring on selected days of the week. In analyzing the personal car data, we have extracted and clustered the positions of stops lasting 30 minutes or more. Besides two clusters near person's home and work, three additional clusters have been detected. The time histograms in Fig.2 show the daily distribution of the stops of these clusters on Saturday and Sunday (A) and from Monday to Friday (B). By means of the interactive filtering tool, we have filtered out clusters 1 and 2 as well as noise. We have also applied histogram zooming for a more convenient consideration of the remaining clusters. It may be seen that the stops of clusters 3 (green) and 4 (orange) occur mostly in the morning and at midday (from 10 to 14 o'clock) on weekends (Fig.2A) and in the evening on working days (Fig.2B); the maximum is attained from 18 to 19 o'clock.

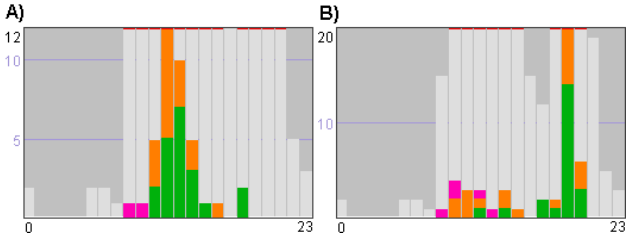


**Figure 2. The daily distribution of three selected clusters of stops on weekends (A) and on working days (B).**

The observed peculiarities of the temporal distribution evoke a hypothesis that the stops of clusters 3 and 4 may be related to shopping. The consideration of the positions in Google Earth supports this hypothesis. Cluster 5 (purple) contains quite a few positions, and their temporal distribution does not allow us to guess the purpose of the stops. In Google Earth, the place looks like a tennis court, but we are not fully confident.

In the truck data, we could not detect any suggestive patterns in the temporal distribution of the stops. With the help of Google Earth, we could confidently recognize only two places of frequent stops lasting 3 hours or more; we shall further call them truck depots. In fact, the lack of domain knowledge does not allow us to choose the right time threshold for the extraction of significant places. In particular, we do not know how much time is required for loading and unloading of a truck. However, we suspect that these times may be quite close to times spent in traffic jams or at stations for collecting road tolls. Hence, the approach based on time thresholds may have quite a limited applicability in this case.

## 4.3 Extraction of Trips

As we have noted, the movement of an entity is represented in the data by merely a sequence of position records. This sequence

needs to be partitioned into subsequences corresponding to trips. The notion of trip may be application- and goal-dependent; therefore, out toolkit for analysis of movement data allows the users to divide data in different ways.

When 'trip' means getting from one significant place to another, it is reasonable to divide the data into subsequences of positions between significant places, which need to be identified earlier. However, the desired level of refinement may vary. Thus, in analyzing the personal car data, the analyst may wish to consider trips from work to home with intermediate stops at one of the shopping centers. In this case, it would be inappropriate to split the data into trips from work to the shopping centers and from the shopping centers to home.

One of the possible ways of dividing movement data into trips, which works quite well for the personal car data, is based on the use of a time threshold, similarly to the extraction of the positions of stops. When the threshold is large, such as 3 hours, the stops at the shopping centers on the way from work to home will be included in the trips from work to home. Using a smaller threshold, e.g. 15 minutes, will result in smaller trips where the shopping centers are destinations or sources.

In analyzing seasonal migration of animals, 'trip' may mean the movement of an animal during the entire migration period, e.g. the movement of a white stork from Europe to Africa in autumn and returning back in spring. Our tool for the extraction of trips allows the user to divide movement data according to temporal cycles. The user needs to choose the appropriate cycle (yearly, weekly, or daily) and one or more positions in this cycle (dates in a year, days in a week, or times in a day) to be used as dividers. Thus, the analyst of the stork migration data could select the yearly cycle and July 1 as the divider.

Both ways of dividing movement data into trips are implemented as database queries, which are hidden from the user behind a friendly user interface. The result of a query is a new database table containing the same data as the original table with an addition of a field with trip identifiers. All records belonging to the same trip will have one and the same trip identifier. Now, the trips can be visualized on a map display. The visualization of trips is discussed in the next section.

With the truck data, none of the two division methods works sufficiently well. There is a need to combine several approaches. From the statistics of distances between successive positions of the same truck we learn that there are a few very high values (the maximum is 23.62 km) indicating lapses in position recording. Our division methods include a method for division by a spatial gap, i.e. the distance between successive positions exceeding a given threshold. In the truck dataset, the average distance between positions is 0.21 km, and all but 28 distances are below 1 km. Hence, 1 km is a suitable threshold for dividing the data.

After this preliminary division, which gives us 71 subsequences, we can refine them further using the two places we have earlier identified as truck depots. It is quite realistic to assume that the trucks often started their trips from these places. Hence, it is reasonable to apply the following method of division: a sequence of positions is split into two parts as it passes one of specified places. The places to use for the division must be defined as area objects. Such objects can be built manually, e.g. by outlining a cluster of points, or automatically by building circles around selected clusters. An additional parameter of the method is the minimum time spent in a place: when the time is below this threshold, no split is made. Dividing the truck data by the place-based partitioning method using the two places of the truck depots and the time threshold of 300 seconds gives us 981 subsequences, of which 537 may be interpreted as round trips from one of the depots and 347 as round trips from the other depot. There are also 8 trips between the depots (4 trips in each direction).

Hence, our toolkit includes four methods for dividing movement data into trips: by temporal gap, by temporal cycles, by spatial gap, and by specified places. It should be noted that the extraction of trips is not done once and forever. The analyst may try various methods and/or parameters of division. However, the results of divisions are stored in the database and may be reused later.

When divided data are loaded from the database into the visual analytics system, each subsequence of records belonging to one trip (i.e. having a common trip identifier) forms a single object. In the process of loading, a number of attributes are extracted or computed and associated with the trip objects: entity identifier, number of positions, duration, date and time of the trip start and end, and positions in relevant temporal cycles, i.e. month of a year, day of a week, and/or hour of a day.

## 4.4 Examination of Trips

### *4.4.1 Viewing Individual Trips*

Individual trips are shown on a map display as lines with specially marked starts (small hollow squares) and ends (bigger filled squares); see Fig.3. When multiple trips are displayed, the lines often heavily overlap, which makes the view illegible. Hence, this way of presenting trips can only be used in combination with interactive filtering tools, such as the time filter (Fig.4). The user selects a time interval, and all displays in the system show only data from this time interval. When a trip does not fully fit in the selected interval, the appropriate part is shown. The user may choose a convenient temporal granularity, which may range from seconds to years depending on the time span of the data. The time filter can also be used as a device for controlling display animation. The user can either drag the slider (blue bar in Fig.4) representing the selected time interval or use the buttons.

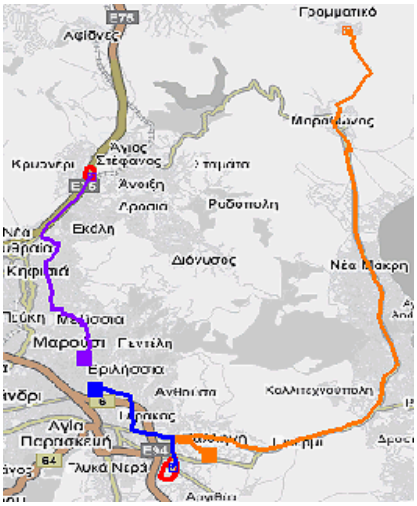


**Figure 3. The map shows three selected trips of different trucks. The red outlines mark the places of the depots.**

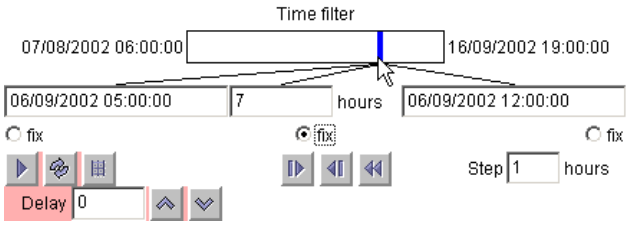


**Figure 4. The interface of the interactive time filter.**

Besides the time filter, there is an interactive attribute filter for filtering data on the basis of attribute values. Thus, trips can be filtered according to their duration or the entities that moved. It is also possible to use the attribute filter for the selection of trips according to days of a week or times of a day.

With the help of Google Earth, selected individual trips may also be displayed as three-dimensional lines in a perspective view with the vertical dimension representing time, either absolute or relative to the start or end of each trip. This is similar to the representation of trips in a space-time cube [8][10][11].

The tools supporting the consideration of selected individual trips are necessary but not sufficient. It is also essential to support an overall view of the whole set of trips and comparison between subsets of trips. A suitable approach is based on clustering, i.e. grouping trips by similarity and consideration of the groups.

### *4.4.2 Clustering of Trips*

Trips may be similar in different respects: they may fully or partly coincide in space, or just have similar shapes, or have common starts and/or ends; they may be fully or partly synchronous or disjoint in time but have similar dynamic behaviors. It depends on the application and goals of analysis which of these respects are relevant. Therefore, it is useful to have a clustering tool allowing the analyst to choose an appropriate similarity measure (also called distance function) from a number of alternatives. We have already mentioned the clustering algorithm OPTICS [4], which we used for the clustering of stop positions. In fact, it is not much important which particular clustering algorithm to use. The main idea is to implement the algorithm in such a way that cluster building is separated from the computation of distances and neighborhood. As a result, the same algorithm can not only be used for the clustering of either points or trips but also for the clustering of trips on the basis of various distance functions.

According to our framework, the toolkit for movement analysis should provide a range of distance functions and allow extension with new functions. Appropriate distance functions are described e.g. in [16][19]. Here we shall demonstrate the benefits of the complementary use of several distance functions even when some of the functions are quite simple. Irrespective of the function chosen, the clustering algorithm uses two parameters: the maximum allowed distance between neighboring objects in a cluster, which is chosen depending on the application, and the minimum size of a cluster. Distance functions may additionally have their specific parameters. In the following examples, we shall use 3 as the minimum cluster size.

In analyzing trips, it is often useful to group them by spatial closeness of their starts and ends. Adequate for this purpose is the function "common start and end" returning the sum of the distances between the starts and the ends of two trajectories divided by two. Let us apply it to the trips of the personal car defined by temporal gaps of 3 hours or more. With the distance threshold of 1000m, the tool builds three clusters. Cluster 1 consists of 110 trips starting around the place of work and ending around home (Fig.5), cluster 2 is formed by 111 trips from around home to work, and cluster 3 includes 52 round trips starting and ending near home. The remaining 17 trips, which have unusual starts and/or ends, are classified as "noise". The examination of clusters is supported by one more interactive filtering tool, which allows the user to switch the clusters on or off (top left of Fig.5).

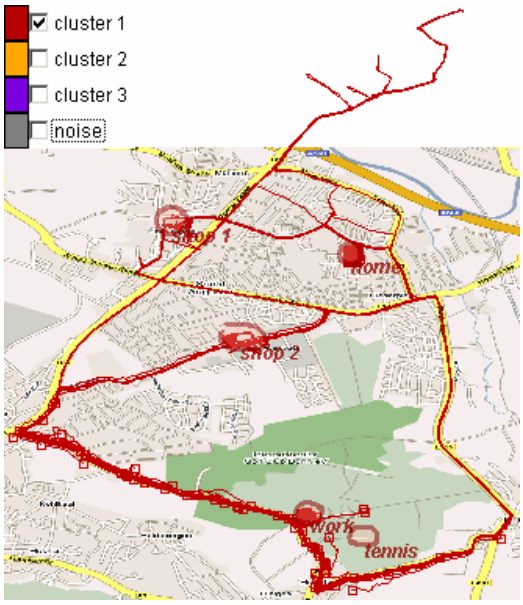


**Figure 5. A cluster of trips of the car from work to home.**

Fig.5 reveals some peculiarities of the personal car dataset. It may be noted that many trips start on streets rather than at work. The reason is that the initialization of the GPS device and the search for satellites takes time, and the recording of the car positions may begin later than the car starts a trip. Positioning errors can also be seen: some trips seem to start in a forest or in a field. These features of the data made us use the quite big distance threshold.

An important property of the clustering tool is that it takes into account data filtering and builds clusters from the currently visible subset of data rather than the whole set. This property can be utilized, in particular, for the kind of analysis that may be called "progressive clustering": the analyst selects one or a few clusters and refines them by re-applying the clustering tool with a different distance function or different parameter settings.

Thus, Fig.6 shows one of possible refinements of the cluster of trips from work to home. The trips have been divided into smaller clusters according to the routes taken. For this purpose, we have used the distance function "common route", which has been devised so that it can tolerate incomplete trip data. The algorithm of the function is presented in Fig.7. The idea is that two trajectories are repeatedly scanned in search for the closest pair of positions. In the course of scanning, two derivative distances are computed: the mean distance between the corresponding positions and a penalty distance. Skipping a position increases the penalty distance (lines 8 and 13). Finding corresponding positions decreases the penalty distance (line 18). The final result is the

sum of the two derivative distances. The clusters presented in Fig.6 have been produced using the distance threshold of 250m.

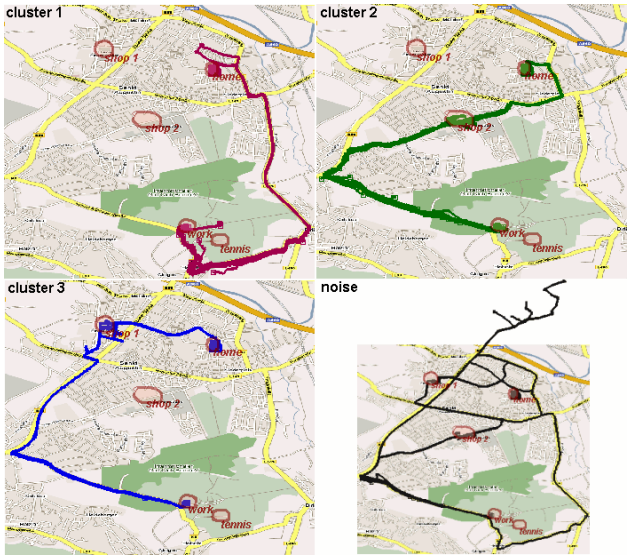


**Figure 6. The trips from work to home have been clustered according to the routes taken.**

```
Algorithm: "common route" distance function
Input: trajectories P, Q; distance threshold D
Output: distance between P and Q
1. dist = 0; pen = 0      //distance and penalty
2. n = 0;       //number of corresponding points
3. i = 1; j = 1   //indices of points in P and Q
4. WHILE i <= P.length AND j <= Q.length
5.    d = point_distance(Pi,Qj)
6.    WHILE i+1 <= P.length AND
7.          point_distance(Pi+1,Qj) < d
8.      pen = pen + point_distance(Pi,Pi+1)
9.      i = i+1; d = point_distance(Pi,Qj)
10.   END WHILE
11.   WHILE j+1 <= Q.length AND
12.         point_distance(Pi,Qj+1) < d
13.     pen = pen + point_distance(Qj,Qj+1)
14.     j = j+1; d = point_distance(Pi,Qj)
15.   END WHILE
16.   dist = dist + d; n = n + 1
17.   IF dist / n > D THEN RETURN D*2 END IF
18.   pen = pen - (D - d)
19.   i = i + 1; j = j + 1
20. END WHILE
21. dist = dist / n
22. WHILE i <= P.length
23.   pen = pen + point_distance(Pi-1,Pi)
24. END WHILE
25. WHILE j <= Q.length
26.   pen = pen + point_distance(Qj-1,Qj)
27. END WHILE
28. RETURN dist + pen
```

**Figure 7. The algorithm of the "route similarity" distance function.**

Both distance functions introduced so far ignore the temporal aspect of the trips. There is a variant of the "common route" function, called "common route and dynamics", which takes this aspect into account in the search for corresponding positions. It begins with finding the first closest pair of positions, to tolerate possible incompleteness of the trajectories, but the following corresponding positions are defined by adding a user-specified time step to the times of the previous positions. Hence, the function groups together trips with close routes and similar dynamics, as illustrated in Fig.8. Here, we have applied the distance function "common route and dynamics" to the trips of cluster 1 in Fig.6 using the distance threshold of 80m and the time step of 3 seconds. As a result, the original cluster consisting of 60 trips has been divided into two sub-clusters containing 18 and 5 trips, respectively, and "noise". Fig.8 shows the two sub-clusters. The 3D view has been rotated so that the left and right sides of the image correspond to south and north, respectively.

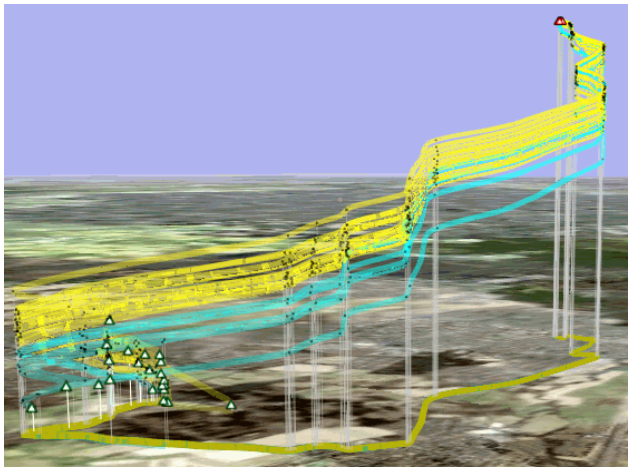

**Figure 8. Two clusters of trips follow the same route but differ in the dynamics.**

Here, the vertical dimension represents the time relative to the ends of the trips: the ends of the trips have the same vertical position and the remaining points of each line are shifted down in relation to this position proportionally to the temporal distance to the end of the trip. The lines in yellow and cyan represent the trips of the first and second sub-clusters, respectively. On the left and in the centre of the image, the vertical positions of the yellow lines are higher than those of the cyan-blue lines. This means that the trips of the first cluster were shorter in time than the trips of the second cluster. The cyan-blue lines are steeper than the yellow ones, which signifies that the speeds in the second cluster of trips were lower than in the first cluster.

With the truck data, clustering helps us to check whether the two places we have detected are really depots or stations to which all trucks repeatedly come. Earlier, we have divided the data into 981 trips by these two places. If our hypothesis about their role is right, all the trips must either start or end at one of them. To check this, we apply clustering with the distance function "common start and end" and the distance threshold of 1000m. As a result, we obtain a cluster of 347 round trips from depot 1, a cluster of 537 round trips from depot 2, and two clusters with 4 trips in each of them starting at one of the depots and ending at the other.

Now, we focus on the remaining 89 trips and again apply clustering using this time the distance function "common start", which returns the distances between the starting positions of the trips. As a result, the subset of 89 trips is divided into 37 trips starting at depot 2, 10 trips starting at depot 1, and 42 trips (two clusters and "noise") starting elsewhere. Once more, we apply clustering to these 42 trips using the distance function "common

end" returning the distances between the end positions of the trips. The result is a cluster of 33 trips ending at depot 2, a cluster of 6 trips ending at depot 1, and three remaining trips, or "noise".

Hence, all but 3 trips have their starts and/or ends at depot 1 and/or depot 2, and most of them are round trips. This strongly supports the hypothesis about the role of these two places. However, there are three trips that do not visit either of the depots. Two of them have a common start and the third trip is spatially separated from these two. This result indicates that there are at least two additional significant places, which we could not identify with the help of Google Earth. A probable reason is that the data, which were collected in 2002, are older than the images in Google Earth. With the use of the map display, we examine in detail the places near the starts and ends of the 3 residual trips and find that the areas around the common start of two trips and around the end of the third trip were also often visited by other trucks and, hence, may be significant. We outline these areas and use them to further divide the trips. This gives us 1075 trips. By means of clustering, we can now examine this new set of trips. Thus, we can learn that 24 trips were made from depot 1 to one of the places and 23 trips from this place to depot 1; 63 trips from depot 2 to the other place and 60 trips in the opposite direction.

In this section, we have demonstrated the use of the clustering tool. A key feature of it is the possibility to use various distance functions, depending on the goals of analysis. This also enables the analytical procedure of progressive clustering where clusters once obtained are refined through further clustering.

### *4.4.3 Summarization of Trips*

The representation of multiple trips by lines does not allow the analyst to see how many trips are there and to distinguish frequent paths from less frequent and occasional ones. We have devised a method for representing multiple trips in a generalized and summarized way (Fig.9): arrows (vectors) show the movement directions; the thickness is proportional to the number of moves.

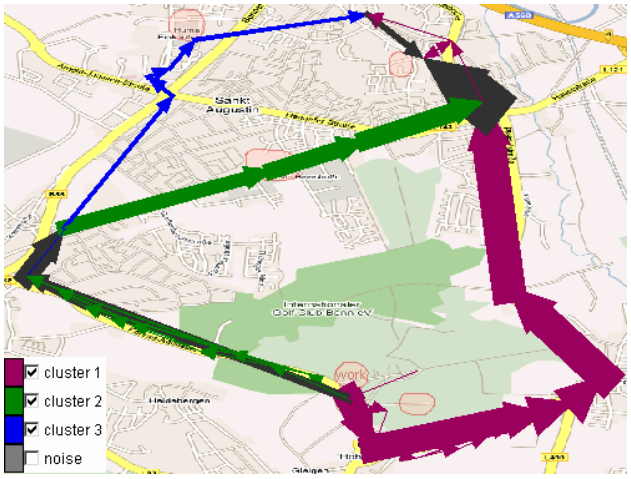


**Figure 9. Summarized representation of trips: clusters 1-3 from Fig.7.**

The algorithm of the summarization is given in Fig.10. It comprises three major steps: extraction of characteristic points of the trajectories, i.e. starts, ends, stops, and turns (lines 1-12); generalization from points to areas by building circles around the points (lines 13-37); and collecting moves (fragments of the trajectories) between pairs of circles (lines 38-58). From the resulting set of aggregate moves, a new map layer is built.

```
Algorithm: summarization of moves
Input: set of trajectories S; minimum angle A;
minimum stop duration D; distance tolerance ε;
minimum radius R0; maximum radius R
Output: set of aggregate moves (flows)
1.  //Step 1: extract characteristic points
2.  P = ∅  //set of characteristic points
3.  FOR EACH trajectory T ∈ S
4.    P = P ∪ {T.first, T.last}
5.    FOR k = 2 to T.length - 1
6.      IF (Tk+1.time - Tk.time >= D AND
7.          point_distance(Tk, Tk+1) < ε) OR
8.         angle(Tk-1,Tk,Tk+1) >= A THEN
9.        P = P ∪ {Tk}
10.     END IF
11.   END FOR
12. END FOR
13. //Step 2: build circles around the points
14. C = ∅  //set of circles
15. WHILE P ≠ ∅
16.   take point p ∈ P
17.   c = new circle(c.center.x = p.x,
18.            c.center.y = p.y, c.radius = R0)
19.   C = C ∪ {c}
20.   xmin=xmax=p.x; ymin=ymax=p.y
21.   Pc = {p} //set of points fitting in c
22.   prev_size = P.size; P = P \ {p}
23.   WHILE P ≠ ∅ AND P.size < prev_size
24.     prev_size = P.size
25.     FOR EACH point q ∈ P
26.       IF inside(q,c) THEN
27.         Pc = Pc ∪ {q}; P = P \ {q}
28.         xmin=min(xmin,q.x); xmax=max(xmax,q.x)
29.         ymin=min(ymin,q.y); ymax=max(ymax,q.y)
30.         c.center.x = (xmin+xmax)/2
31.         c.center.y = (ymin+ymax)/2
32.         c.radius =
33.             min(R, R0+ max(xmax-xmin,ymax-ymin)/2)
34.       END IF
35.     END FOR
36.   END WHILE
37. END WHILE
38. //Step 3: summarize moves between the circles
39. M = ∅  //set of aggregate moves
40. FOR EACH trajectory T ∈ S
41.   C0 = find(c0 ∈ C: inside(T.first,c0)); i = 1
42.   FOR k = 2 to T.length
43.     IF NOT inside(Tk,c0) THEN
44.       c1 = find (c1 ∈ C: inside(Tk,c1))
45.       IF c1 ≠ null THEN
46.         m = find(m∈M: m.start=c0 AND m.end=c1)
47.         IF m = null THEN
48.           m = new aggregate_move(m.start = c0,
49.                    m.end = c1, m.moves = ∅)
50.           M = M ∪ {m}
51.         END IF
52.         m.moves = m.moves ∪ [Ti,Tk]
53.         c0 = c1; i = k
54.       END IF
55.     ELSE i = i+1
56.     END IF
57.   END FOR
58. END FOR
59. RETURN M
```

**Figure 10. The algorithm of the summarization of moves.**

Note that an aggregate move is an object containing references to the original moves it unites (line 52). This allows the object to react to any filter applied to the trajectories. When the user selects a subset of trajectories or a time sub-interval, each aggregate move checks which original moves satisfy the filter conditions. In absence of such moves, the aggregate move does not appear on the map; otherwise, the thickness of the vector is adjusted to the number of the filter-compliant original moves. If all these moves belong to trajectories of the same cluster, the vector is colored in the color of the cluster; otherwise, it is shown in gray.

The 2D summarized view is well suited for simple routes without returns and intersections but lacks clarity in more complex cases. In the future, we plan to implement a 3D view where the vectors are positioned in the vertical dimension according to the relative times of the corresponding moves. This will decrease the intersections and overlapping of the vectors.

Besides the spatial summarization, non-spatial aggregation of trip characteristics may provide additional insight concerning the movement. Thus, the temporal histograms in Figure 11 portray the distribution of the start times of trips with respect to the weekly and daily cycles. As in Figs.1 and 2, trip clusters and their colors have been transmitted to the histograms. Fig.11A shows how the three major routes of the personal car from work to home are distributed over a week. The eastern route (red) notably prevails on Fridays. On Thursdays, the routes via shops 1 and 2 (blue and green) are chosen as frequently as the eastern route. Curiously, the route via shop 1 has never been taken on Tuesday.

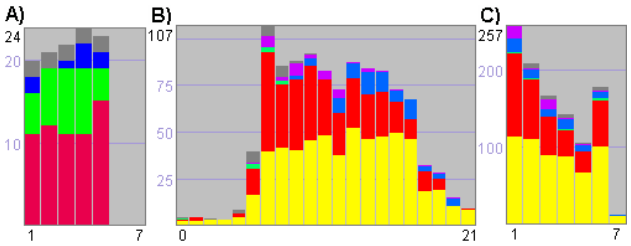


**Figure 11. Temporal histograms show the distribution of the start times in trip clusters over a week (A,C) and a day (B).**

Fig.11B,C show the daily (B) and weekly (C) distributions of the start times of the trips of the trucks, which have been clustered according to the closeness of their start positions. Clear differences can be seen between the distribution patterns of the trips originating from depot 1 (red) and depot 2 (yellow). Blue and purple correspond to trips from two additionally detected significant places. The former tend to occur mostly at midday and in the afternoon, the latter show the opposite tendency.

### *4.4.4 Examination of Visited Places*

In the course of summarization, not only the moves between the generalized places (circles) are collected but also various statistics for the places computed: numbers of visits, of different trips, of starts and ends; minimum, maximum, mean, and median speeds, times spent, turns, etc. Any of these statistics can be visualized. Like the numbers of moves, the place visits statistics are dynamically re-computed in response to changes of the data filtering, and the visualization is automatically updated.

Fig.12 presents the minimum and median times spent in different places by the trucks in the trips originating from depot 1. Filtering has been applied to the places: shown are only the places that were visited at least twice and the minimum time spent was at least 5 minutes. This display allows us to detect the likely destinations of the trips. An analyst who knows the destinations (e.g. a logistic manager from the truck company) could use this display to detect places where much time is lost. The analyst might also be interested in looking at the speeds statistics or other computed attributes.

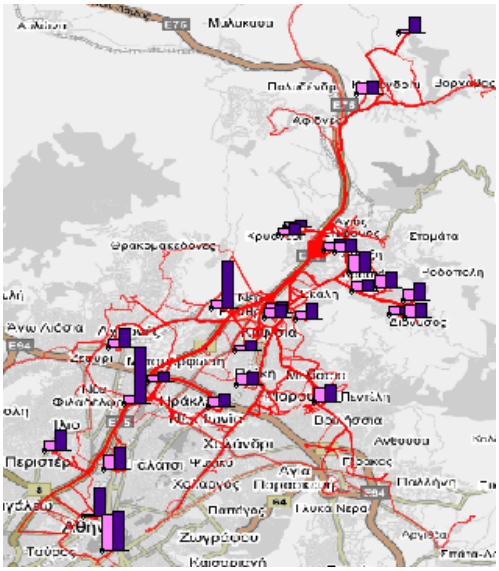

**Figure 12. The bar charts show the minimum and median times spent in different places during the trips of the trucks originating from depot 1.**

## 5. POTENTIAL APPLICATIONS

The purpose of visual analytics techniques is to help a human analyst to understand some data and underlying phenomena. The visual analytics framework we have introduced in this paper is applicable to diverse movement data. Thus, it can be used in studies of individual movement behaviors, including behaviors of animals. It can also be used to analyze movements of multiple entities for the purposes of city planning, traffic management, logistics, optimization of layouts of public venues and shopping areas, allocation of facilities or advertisements, and many others.

At the same time, visual analytics methods are not suitable for applications where movement data are used for personalized real-time services. Such applications require fully automatic methods of data processing as, for example, described in [13][15]. However, visual analytics may be helpful on the stage when such automatic methods are developed and tested.

## 6. CONCLUSION

By example of two different datasets, we have introduced a framework for enabling a human analyst to make sense from large amounts of movement data initially lacking any semantics. In the process of analysis, the meaning appears as the analyst perceives information, links it to his/her prior knowledge and evidence from other sources, and reasons about it. Interactive visual displays play the key role in supporting this process of sense-making but are insufficient for analysis of large and/or complex data. Therefore, besides interactive displays, the framework includes database operations and computations so that the techniques are

complementary and mutually reinforcing. The generic database techniques enable handling large datasets and are used for basic data processing and extraction of relevant objects and features. The computational techniques, which are specially devised for movement data, aggregate and summarize these objects and features and thereby enable the visualization of large amounts of information. The visualization enables human cognition and reasoning, which, in turn, direct and control the further analysis by means of the database, computational, and visual techniques. Interactive visual interfaces embrace all the tools.

## 7. ACKNOWLEDGMENTS

The work has been partly funded by EU in the project GeoPKDD - Geographic Privacy-aware Knowledge Discovery and Delivery (IST-6FP-014915; http://www.geopkdd.eu). We are grateful to our project partner Salvo Rinzivillo (Univ. of Pisa) who actively participated in the development of the progressive clustering tool.

## 8. REFERENCES

[1] Andrienko, N., Andrienko, G., and Gatalsky, P. Supporting Visual Exploration of Object Movement. In V. Di Gesù, S. Levialdi, L. Tarantino (eds.) Proc. Working Conf. Advanced Visual Interfaces (AVI 2000), Palermo, Italy, 2000, ACM Press, 217-220, 315

[2] Andrienko, N., Andrienko, G., and Gatalsky, P. Impact of data and task characteristics on design of spatio-temporal data visualization tools. In Exploring Geovisualization. (Eds: Dykes, J.A., Kraak, M.J., and MacEachren, A.M.) Elsevier, London, 2005, 201-222

[3] Andrienko, N., and Andrienko, G. Designing visual analytics methods for massive collections of movement data. Cartographica, v.42 (2), Summer 2007, 117-138

[4] Ankerst, M., Breunig, M., Kriegel, H.-P., and Sander, J. OPTICS: Ordering Points to Identify the Clustering Structure. In Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99), 1999, ACM Press, New York, NY, 49-60

[5] Buliung, R.N. and Kanaroglou, P.S. An Exploratory Data Analysis (ESDA) toolkit for the analysis of activity/travel data. Proceedings of ICCSA 2004, LNCS 3044, Springer, Berlin, 1016-1025

[6] Dykes, J. A. and Mountain, D. M. Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications, Computational Statistics and Data Analysis, 43, 2003, 581-603

[7] Forer, P., and Huisman, O. Space, Time and Sequencing: Substitution at the Physical/Virtual Interface. In Information, Place and Cyberspace: Issues in Accessibility (Eds: Janelle, D.G., and Hodge, D.C.), Springer, Berlin, 2000, 73-90

[8] Hägerstrand, T. What about people in regional science? In: *Papers of the Regional Science Association*, 24, 1970, 7-21

[9] Imfeld, S. Time, points and space: Analysis of wildlife data in GIS. Unpublished Dissertation, University of Zürich, Department of Geography, Zürich, 2000, http://www.geo.unizh.ch/~imfeld/diss

[10] Kapler, T. and Wright, W. GeoTime information visualization, Information Visualization, 4(2), 2005, 136-146

[11] Kraak, M.-J. The space-time cube revisited from a geovisualization perspective, in: Proc. 21st Int. Cartographic Conf., Durban, South Africa, 2003, 1988-1995

[12] Laube, P., Imfeld, S., and Weibel, R. Discovering relative motion patterns in groups of moving point objects. International Journal of Geographical Information Science, Vol. 19, No. 6, July 2005, 639–668

[13] Liao, L., Fox, D., and Kautz, H. Hierarchical Conditional Random Fields for GPS-based Activity Recognition. In Results of the 12th Int. Symposium ISRR, Springer Tracts in Advanced Robotics (STAR) 28, Springer, Berlin, 2007, 487-506

[14] Mountain, D.M. Visualizing, querying and summarizing individual spatio-temporal behaviour. In Exploring Geovisualization. (Eds: Dykes, J.A., Kraak, M.J., and MacEachren, A.M.) Elsevier, London, 2005, 181-200

[15] Patterson, D., Liao, L., Gajos, K., Collier, M., Livic, N., Olson, K., Wang, S, Fox, D., and Kautz, H. Opportunity Knocks: a system to provide cognitive assistance with transportation services. In Proc. 6th Int. Conf. Ubiquitous Computing (UbiComp 2004), Nottingham, UK, LNCS 3205, Springer, Berlin, 2004, 433-450

[16] Pelekis, N., Kopanakis, I., Marketos, G., Ntoutsi, I., Andrienko, G., and Theodoridis, Y. Similarity search in trajectory databases. In Proc. 14th Int. Symposium Temporal Representation and Reasoning (TIME 2007), IEEE Computer Society Press, 2007, 129-140

[17] Tobler, W. Experiments in migration mapping by computer, The American Cartographer, 14 (2), 1987, 155-163

[18] Tobler, W. Display and Analysis of Migration Tables, 2005, http://www.geog.ucsb.edu/~tobler/presentations/shows/A_Flow_talk.htm

[19] Trajcevski, G., Ding, H., Scheuermann, P., Tamassia, R., and Vaccaro, D. Dynamic-aware similarity of moving objects trajectories. In Proc. ACMGIS'07, Seattle, USA, 2007, ACM Press, 75-82

[20] Vasiliev, I.R. Mapping Time, Cartographica, **34** (2), 1997, 1-51

[21] Yu, H. Spatial-temporal GIS design for exploring interactions of human activities, Cartography and Geographic Information Science, 33(1), 2006, pp. 3-19

## About the authors:

**Gennady and Natalia Andrienko** are researchers at Fraunhofer Institute IAIS. Their research topics include geovisualization, information visualization, visual data mining, and visual analytics (see http://geoanalytics.net).

**Stefan Wrobel** is the director of Fraunhofer Institute IAIS and a professor at the University of Bonn. His primary research interest is in the area of intelligent analysis and information systems, data mining and machine learning.