Universität Augsburg
Fakultät für Angewandte Informatik

Student Work

Weibo User Activity Analysis based on Spatial Summarization

Geo-Visualization: (WS16)

Supervisor: Dr. Lingfang Ding

Illium, Steffen

Number: 1410069

Geoinformatics Master, fourth Semester

steffen.illium@student.uni-augsburg.de

1 Introduction

The hereafter-presented trajectory aggregation attempt tries to show application of data re-sampling methods. The processed data is later on used to generate understandable visual representations, which deal with the special requirements of spatial and temporal enhanced social media messages.

As a typical representation of any kind of movement through space (and time), a line or a group of consecutive vectors is used. Where a single movement or some distributed lines are visible, bigger datasets tend to show cluttered and over plotted lines. In times of Big-Data-Applications, researchers need to find ways to aggregate and analyze huge datasets with millions of single users and finally represent the results in an intersubjective understandable way. Furthermore, the partial amount of social media messages with a jumping behavior in its spatial context adds another level of abstraction, not just in analysis steps, but also in the datasets themselves.

This work is divided in four chapters, each dealing with the respective smaller problems in building the final visualization. First, statistical methods and in particular clustering will be used to find places of interest (POI) which also represent origin and destination points. After that, Voronoi-Polygons will be used to collect messages and move them to the previously found hot spots. Subsequently per user, trips will be generated from this spatial rearranged dataset. The temporal context will be used for visiting order, hence the shape of a moving entity and its direction. After the re-sampling process of the geometrical structure, a statistical summarization of overlapping trajectories needs to be computed. Finally, the visualization takes its place with the now available, non-overlapping and refined data structure.

2 Structure & Application

The overall workflow is organized as shown in Figure 1 and will be presented gradually in the following sections.

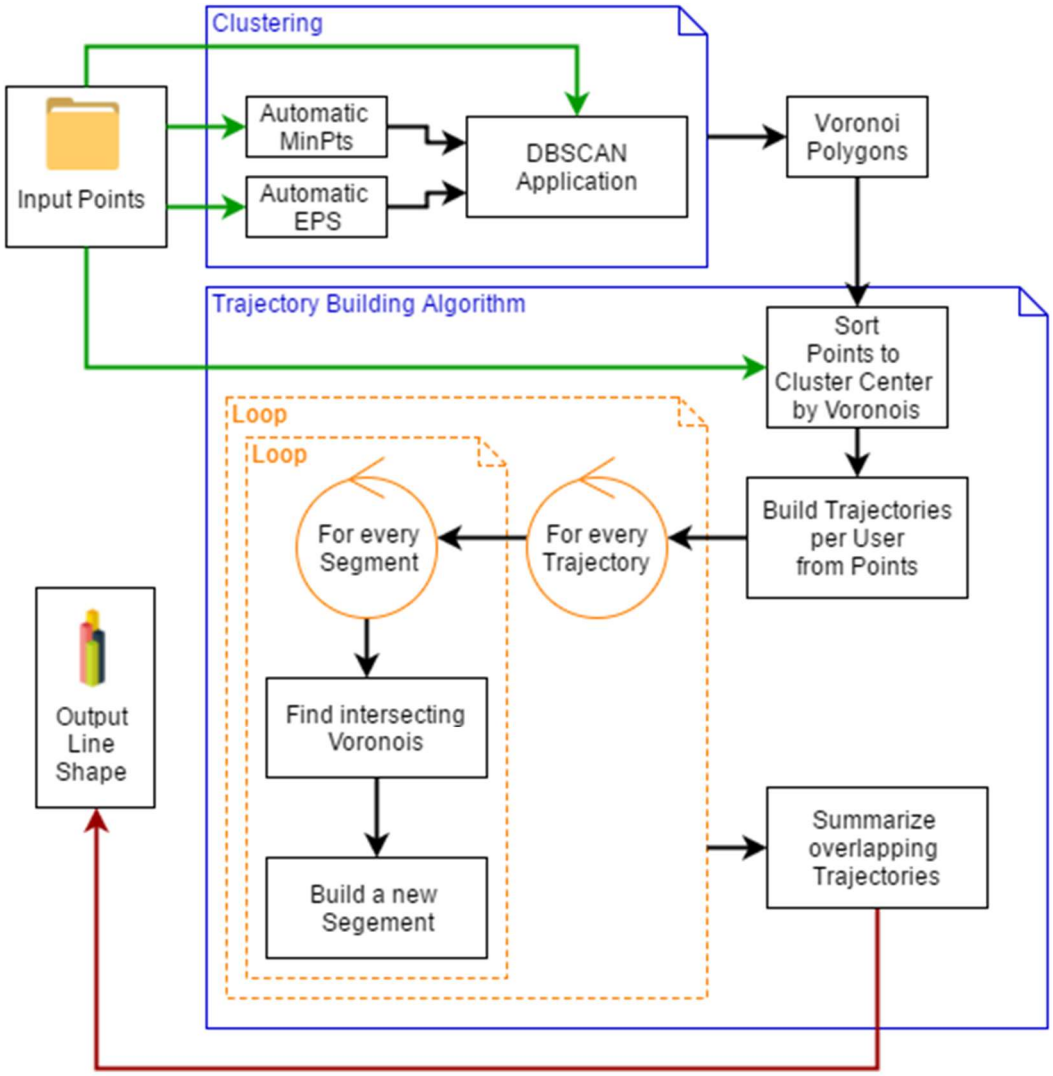


Figure 1: Logical workflow including clustering, trajectory building and re-sampling.

2.1 Clustering

Several approaches in spatial clustering of large datasets such as KMEANS and others (Han, Kamber, & Tung, 2001) already exist. This work however uses a density-based solution of outlier detection and clustering, the widely available "Density-Based Spatial Clustering of Applications with Noise"(DBSCAN) algorithm. Given a search threshold in map units ('eps') and the minimum amount of points of a valid cluster ('MinPts'), DBSCAN calculates the connected spatial groups. Points that are not fitting in given distance into a big enough set of other points are considered noise. (Ester, Kriegel, Sander, & Xu)

In application, a user has to determine 'eps' and 'MinPts' as global density parameters. The algorithm then computes the amount of found clusters and labels corresponding points. A geometric centroid can then be calculated from equally labeled points (Smith, Goodchild, & Longley, 2007, p. 79). These centers are considered as points of interests. However, some of the Weibo users are not directly referring to live-events. Some wait until they found a comfortable situation in which they actually use such a services. That being said,

the here proposed method should be used on data that is more reliable in terms the geospatial context (Siming et al., 2014) or be used in a comfy places context.

As initial step, an automation to retrieve 'MinPts'-Values had to be developed, to achieve a plausible result in terms of the calculated cluster count. Due to the density-based approach of DBSCAN, a density-based approach was also used in this consideration. The area, inclosing all points in km², also known as convex hull (Smith et al., 2007, p. 106), is divided by the total number of all sampling points. This value is considered representative for an area of four km².

$$MinPts = (ConvexhullArea / TotalCount) / 4000$$

The second parameter of DBSCAN should be also retrieved in an automatic and current dimension based procedure. Rahmah & Sitanggang, 2016 developed and introduced such an approach in combination with DBSCAN. The resulting algorithm of their work was based on a generated distance matrix. Since, in this work, a dataset with 90,000 sampling points is used, the procedure would need to compute 8,100,000,000 single distances. Afterwards the lowest three per sampling point would be filtered. This was not possible with the available computing power for this project. As a possible solution, a sample of the initial dataset with about 10,000 data points (reducing the computation to 100,000,000 needed distances) could be used. However, a simple calculation, that takes distances in terms of map units and geographical latitude in consideration, was implemented as a cheap workaround to place the 'eps' in a useable dimension.

$$eps = 0.003 * cos(minLon + (maxLat - minLon)/2)$$

Variations (Figure 2) in the factor by 0.001 need to be tested to achieve a maximized cluster result, since more clusters tend to find groups that have a higher spatial relation. Nevertheless, since visualization is the first topic of this work and re-sampled the second, over plotting and much too fine information would not be beneficial.

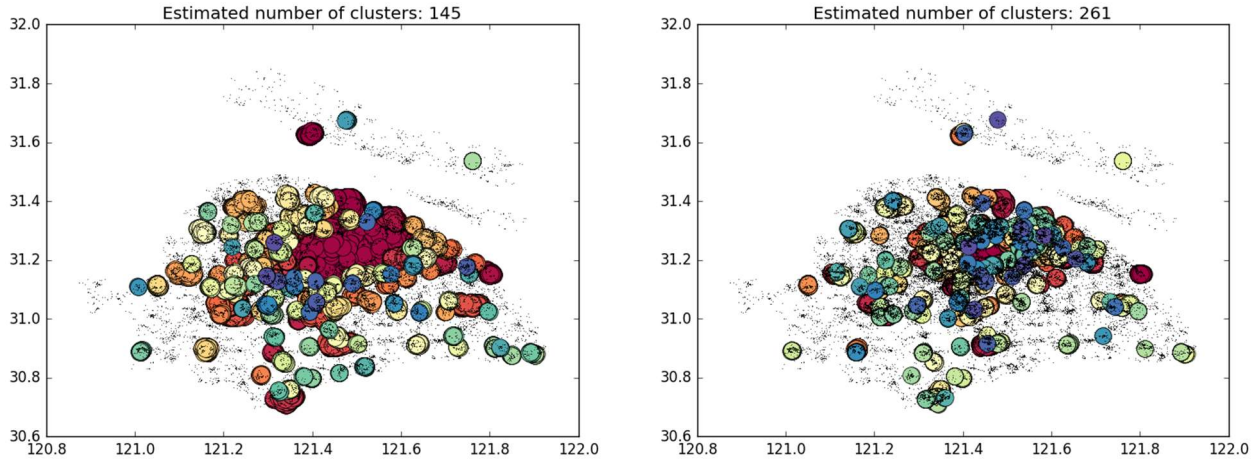


Figure 2: Clustered Weibo Data in Shanghai; left: Clusters: 145, eps: 0.00494 , right: Clusters:261, eps: 0.00296

The hereafter-used cluster count of 261 was retrieved with a factor of 0.003 resulting in an 'eps' of 0.00296 (in Map Units). First results (Figure 2, right) are already showing the center of Shanghai (red eye) with visible suburbs and POI areas in the region surrounding the city.

2.2 Voronoi

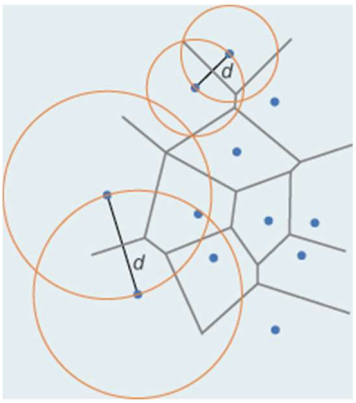


Figure 3: Building Voronoi-Polygons (Haggett, Cliff, and Frey)

In the previous chapter, POIs were generated through the application of DBSCAN clustering and computing of the corresponding centers by calculation centroids of participating points. In this section these centers will be used to form so called Voronoi polygons to gather all related points within its sphere.

Thiessen or Voronoi polygons are a methodical application for neighborhood or proximity analysis in not only but mostly planar Euclidian space (Philipps Universität Marburg). In short, at half distance between two neighboring points, an orthogonal line is drawn until it hits another line or the bounding box surrounding all sampling points. These Voronoi edges (the lines) are building the grid for final Voronoi planes (Aurenhammer, Klein, & Lee, 2013). In the next step, each sampling point of the initial dataset is assumed to belong to its intersecting Voronoi plane. That being said, a discussion of the here used method could be applied, to generate more precise relations.

2.3 Main Process – Sorting

As already mentioned, previously generated Voronoi polygons of cluster centers are now used as a sorting base for all sampling points. The Algorithm alters the initial geographical position of all points to neighboring

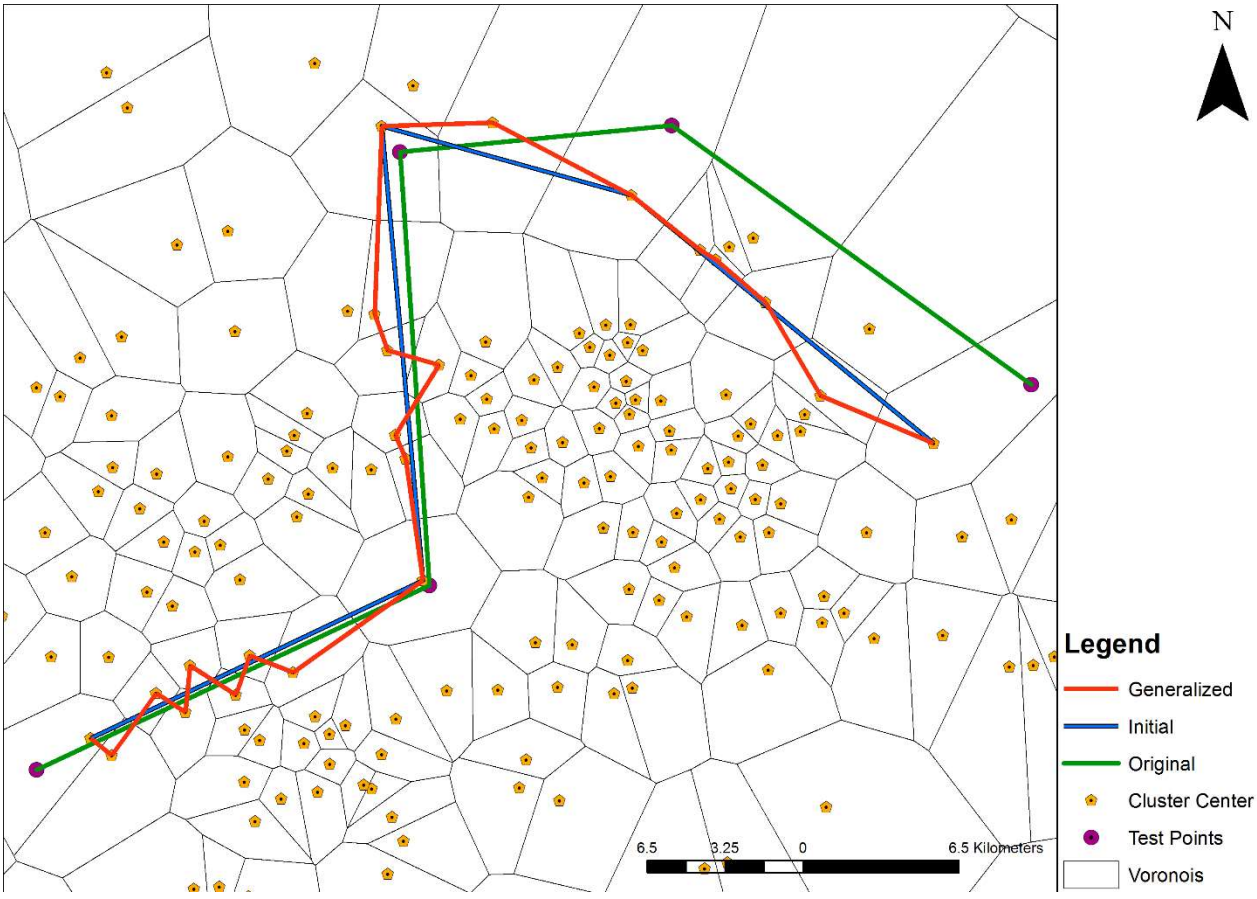


Figure 4: Re-sampled trajectories; Green: The original trajectory, Blue: The initial trajectory after moving data points to cluster center, Red: The re-sampled trajectory only traveling from Voronoi center to neighboring cells.

centers, determined through the proposed method. It should be noted, that not all points that are forming a cluster are inevitably assigned to the Voronoi polygon that is built upon it. The geometrical shape is now disjoint from the associated cluster. Only its overall dimension and position can be preserved. The same behavior applies for the so-called noise, points that could not be fitted within any of the DBSCAN clusters, are now assigned to intersecting Voronoi polygons. As this work focuses on abstraction and re-sampling this is the intended behavior to compute and show an overall trend. Further discussions could lead to better solutions in handling the noise.

2.4 Main Process – Building the trajectories

Now, that every point has been moved, each 'initial' trajectory (Figure 4, green) can be generated per user. A trajectory is assumed to have at least two points, representing a movement through space and time, from start to destination. This can be a single movement from 'a' to 'b' but also a series of movements 'a' to 'b' to 'c'. An overall line of movement is hereafter called a trajectory, where smaller parts, meaning the movements from one points to another will be referenced as segment. So that resulting polylines with a length of zero or less than two participating points (not enough to form at least a segment) are removed within this step.

2.5 Main Process - Splitting to smaller sections

The previously generated 'initial' trajectories already share at least one similarity; each of their segments start and end at Voronoi cluster center. Now, each segment needs to be split so that, no polygon is intersected whose cluster center is not either the start or the endpoint. This step introduces a sorting problem to the algorithm. Determined Voronoi polygons of the intersection check form an unsorted list. Simply connecting their centers would lead to an uncontrollable outcome. The surprisingly most accurate but also a very complex solution would be the implementation of a traveling salesman approach. For the purpose of simplification and processing time, this option was dropped for the favor of an order by distance function.

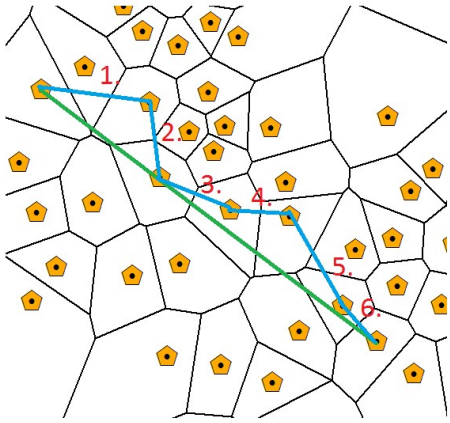


Figure 5: Splitting a segment to smaller sections.

Centers that are very close to the starting point are the first in sequence, followed by the others in ascending order (Figure 4, red, Figure 5, blue). The resulting trajectory indicates an overall trend, not a direct linear movement, due to the abstract nature of the re-sampled pattern. In comparison to Andrienko & Andrienko, 2011 the here presented procedure was not designed to produce summarized trajectories from a lot of small sections, but to produce a lot smaller segments from overall movements.

2.6 Main Process – Statistical summary

Table 1: Attribute meanings

Attribute name	Meaning
userfollow	Number of people the user is followed by
userfriend	Number of people that the user is following
userstatus	Number of messages (statuses) the user posted

The very last step of the proposed algorithm is the statistical summary of overlapping lines. The result of this step should present up to 8 groups, formed by the three main attributes and its variations (Table 2). This is achieved by cross checking the global average with the local average of available attributes. If the average of one attribute is over its global average, a logical 'True' is returned. Furthermore, the amount of overlapping segments is saved for a later usage (labelled as 'size'). The available sample data of Weibo social media network carries three different attributes (Table 1) whose combinations form eight different groups (Table 2). Each set of geometrically similar segments is evaluated and labeled through this operation.

Table 2: Possible combinations that form user groups

	Group 1 'heavy user'	Group 2	Group 3	Group 4	Group 5	Group 6 'reader'	Group 7 'prophet'	Group 8 'partici-pating'
userfollow	True	True	True	True	False	False	False	False
userfriend	True	True	False	False	True	True	False	False
userstatus	True	False	True	False	True	False	True	False

2 Visualization

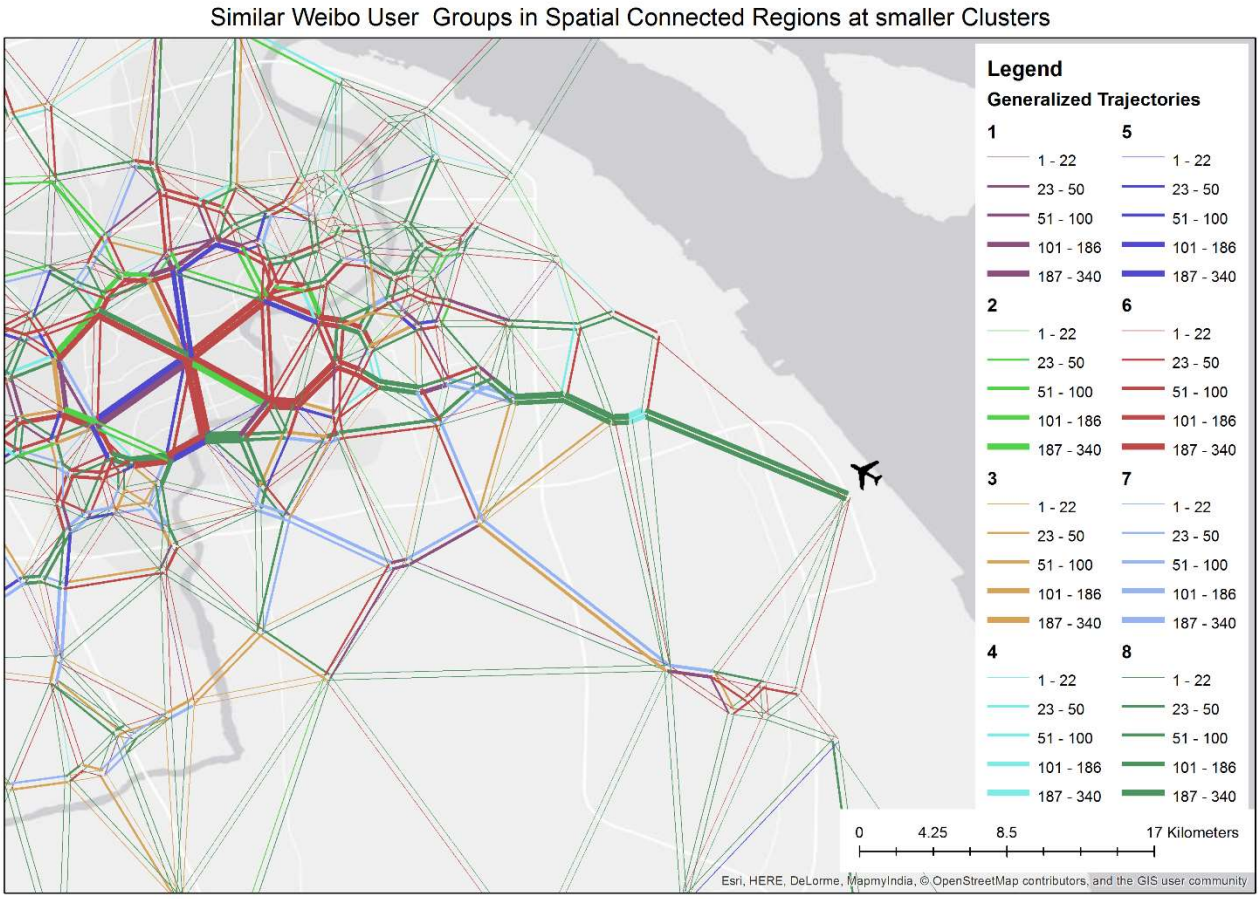


Figure 6: Final visualization of re-sampled Weibo trajectories, colored in labeled Groups, amount of overlapping segments as size

The visualization process including the design of maps and graphs was accomplished by the use of ESRI ArcGIS, although, it was not possible to visualize the trajectories in an appropriate manor. There are major drawbacks in handling lines and its directed movement representation. Due to this fact, it was not possible to implement additional arrows indicating directions.

With a rise in cluster count and single movements in both directions along the axis, again the phenomenon of over plotting and cluttering is visible. On the other hand, while handling an underlying base map, the right color setting and zoom level needs to be chosen. This procedure turned out to be better visible in interactive mode while inspecting specified areas of the city, rather than for full overview. In other words, zoom level needs to have its corresponding level of detail regarding the cluster count.

Nevertheless, promising results could be achieved. Figure 6 shows a zoomed visualization of the presented application. Variations in colors are indicating the different user groups, while variations in line width are presenting the amount of overlapping trajectory segments.

3. Results & Discussion

The here presented workflow tried to aggregate several different movement pattern as general trajectory between city districts. It needed to take care of the jumping behavior of the social media messages on the example of Weibo Social Media Network. At first, clusters within the dataset, based on the spatial location of written messages, were calculated. After that, the overall movement per user was equalized accordingly to

the computed pattern. The final summarization tried to aggregate overlaying movements resulting in segments of similar directions. The overall mean was used to sort those results in similar groups.

As an interesting outcome, the resulting distribution in groups need to be mentioned. The most common one with about 15% (Appendix 3) is Group 6. It neither is writing more messages then the average written messages nor is this group followed by many other Users. However, members of this group are following a more than average number of other users. This group could be named as 'readers'. Its members are often found in the city's center. Another group to mention is Group 8. Users belonging to this group are participating the Weibo service under average in every analyzed attribute. They are found in the big sized trajectory, travelling to the airport as well as in the southern suburbs. By looking at just those two groups (Appendix 2) an additional pattern gets visible, dividing the map in city center and surrounding areas. Variations in cluster size verifies those results, while disrupting the overall movement structure (Appendix 1). Nevertheless, the center of gravity is still to be found at the very same spot.

Overall, this method showed clear results regarding the correlation of spatial close groups (Toblers first law of geography (Miller, 2004)) as well as regarding the preservation of the overall shape of movements and infrastructure. It is advised to work with a slightly larger cluster size than desired. Then, zoomed areas of the resulting image should be visualized. Further, attributes that are more meaningful should be used to achieve an additional information gain. This workflow is an appropriated approach in re-sampling large movement-datasets with a jumping behave on users side.

References

Andrienko, N., & Andrienko, G. (2011). Spatial generalization and aggregation of massive movement data. *IEEE transactions on visualization and computer graphics*, *17*(2), 205–219. doi:10.1109/TVCG.2010.44

Aurenhammer, F., Klein, R., & Lee, D.-T. (2013). *Voronoi diagrams and Delaunay triangulations*. New Jersey, NJ: World Scientific.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.

Han, J., Kamber, M., & Tung, A. K. H. (2001). Spatial Clustering Methods in Data Mining: A Survey. In H. J. Miller & J. Han (Eds.), *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS.* Taylor and Francis. Retrieved from http://www-faculty.cs.uiuc.edu/ hanj/pdf/gkdbk01.pdf

Miller, H. J. (2004). Tobler's First Law and Spatial Analysis. *Annals of the Association of American Geographers*, *94*(2), 284–289. doi:10.1111/j.1467-8306.2004.09402005.x

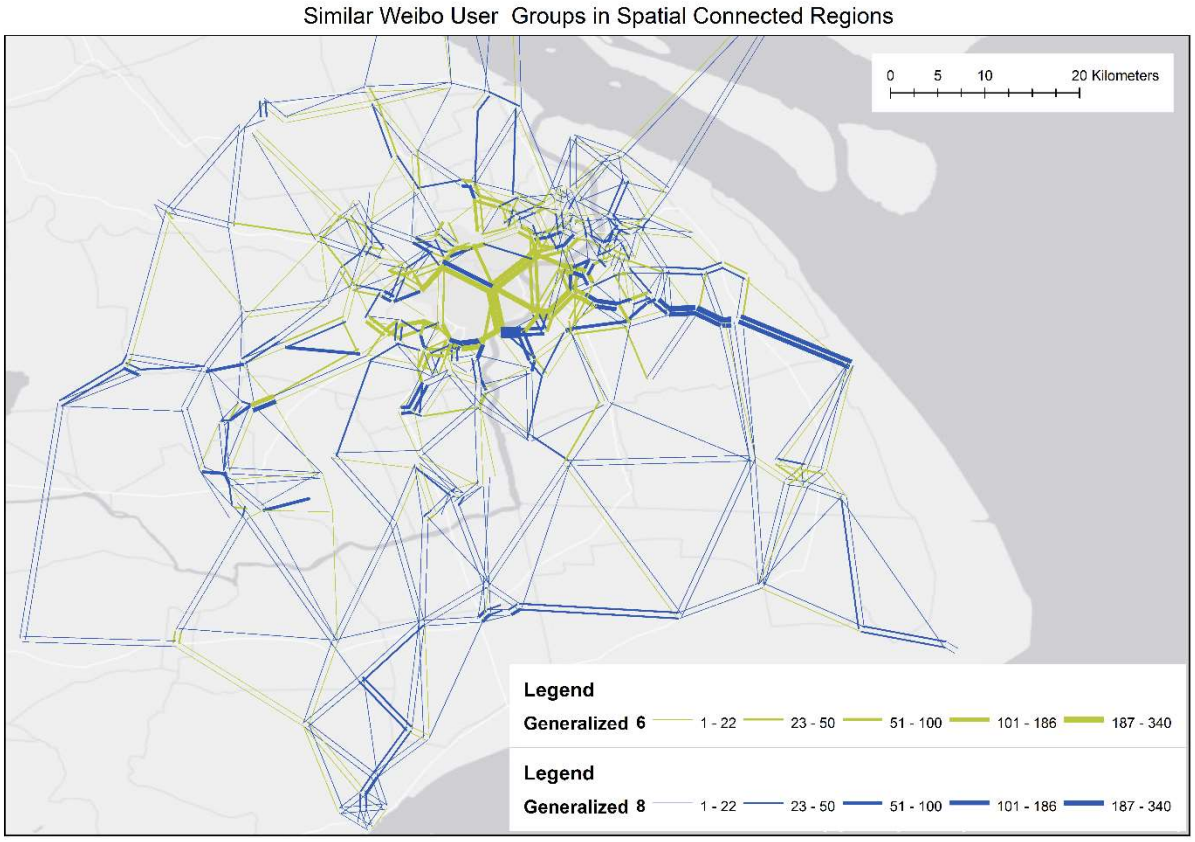
Philipps Universität Marburg. Distanzbeziehugnen: Voronoi Polygone als Ableitung von Entfernung und Nachbarschft. Retrieved from http://gisbsc.gis-ma.org/GISBScL6/de/html/GISBSc_VL6_V_lo5.html

Rahmah, N., & Sitanggang, I. S. (2016). Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra. *IOP Conference Series: Earth and Environmental Science*, *31*, 12012. doi:10.1088/1755-1315/31/1/012012

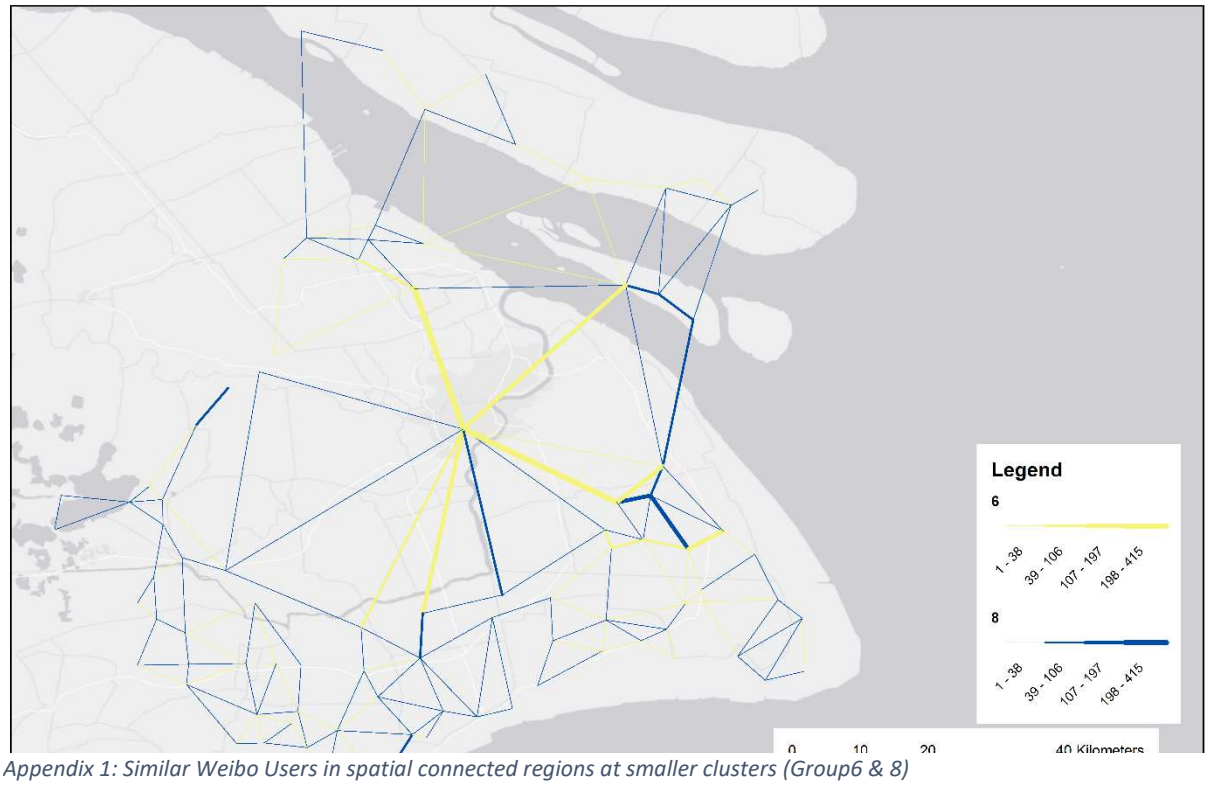
Siming, C., Xiaoru, Y., Zhenhuan, W., Cong, g., Jie, L., Zuchao, W.,. . . Jiawan, Z. (2014). Interactive Visual Discovering of Movement Patterns from Sparsely Sampled Geo-tagged Social Media Data.

Smith, M. J. de, Goodchild, M. F., & Longley, P. A. (2007). *Geospatial analysis: A comprehensive guide to principles, techniques and software tools*. Leicester: Matodor.

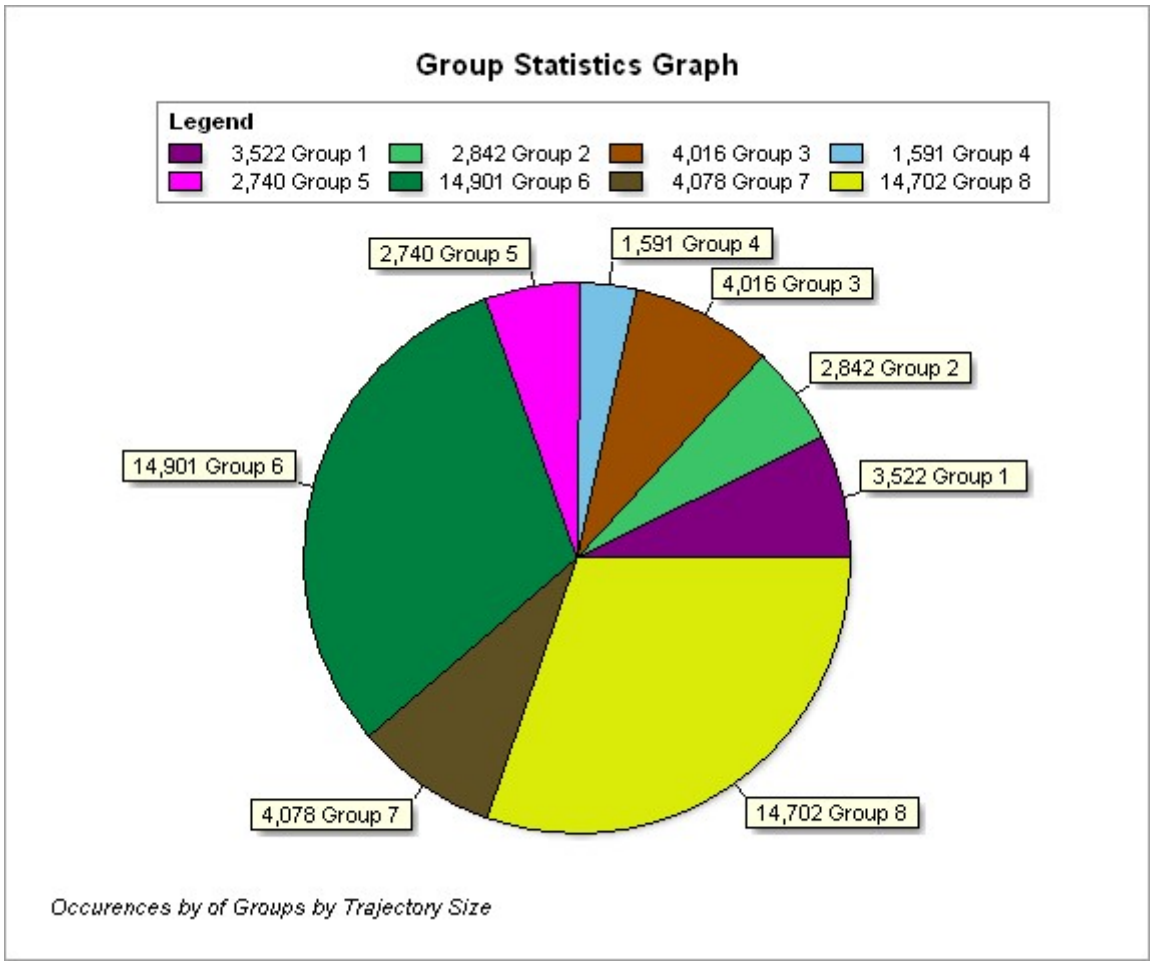
Appendix



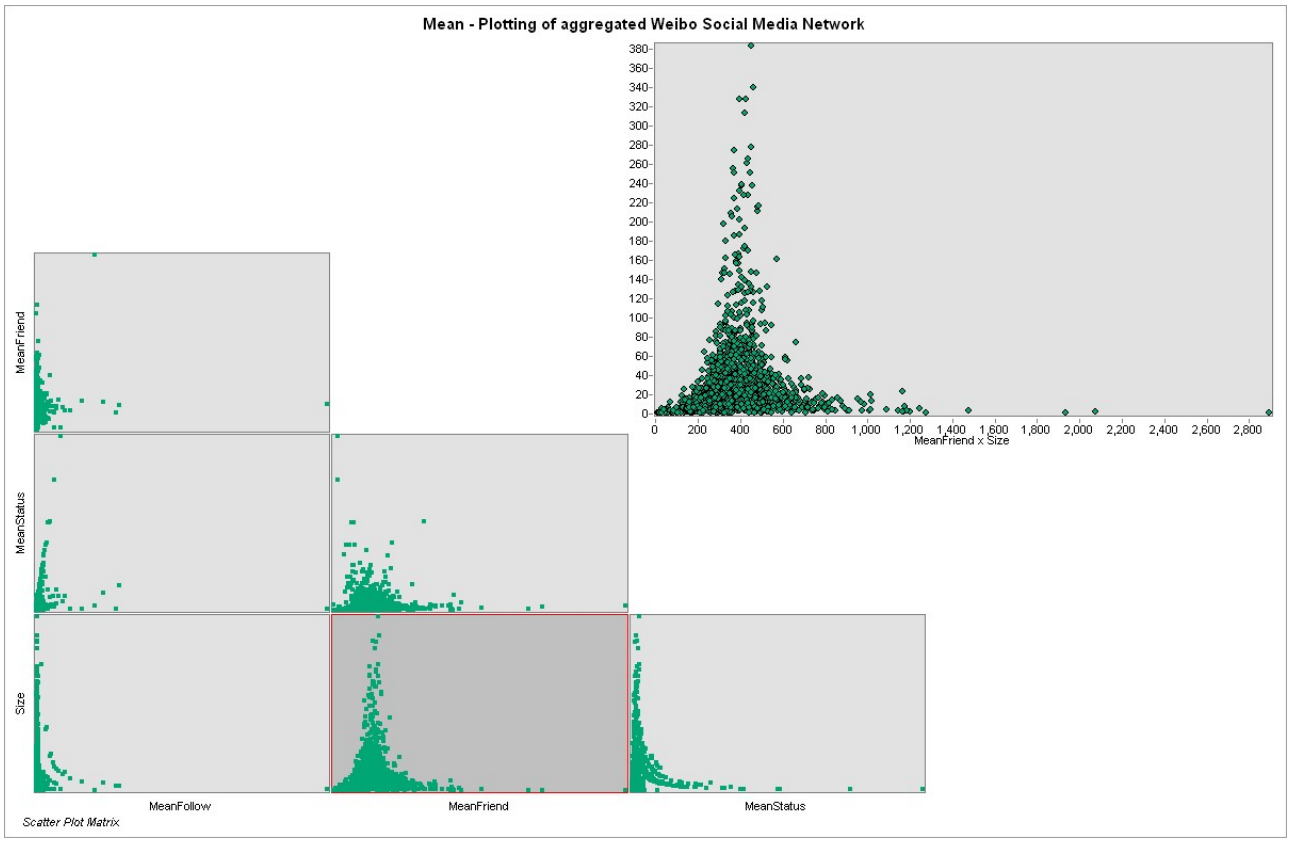
Appendix 2: Similar Weibo Users in spatial connected regions (Group6 & 8)



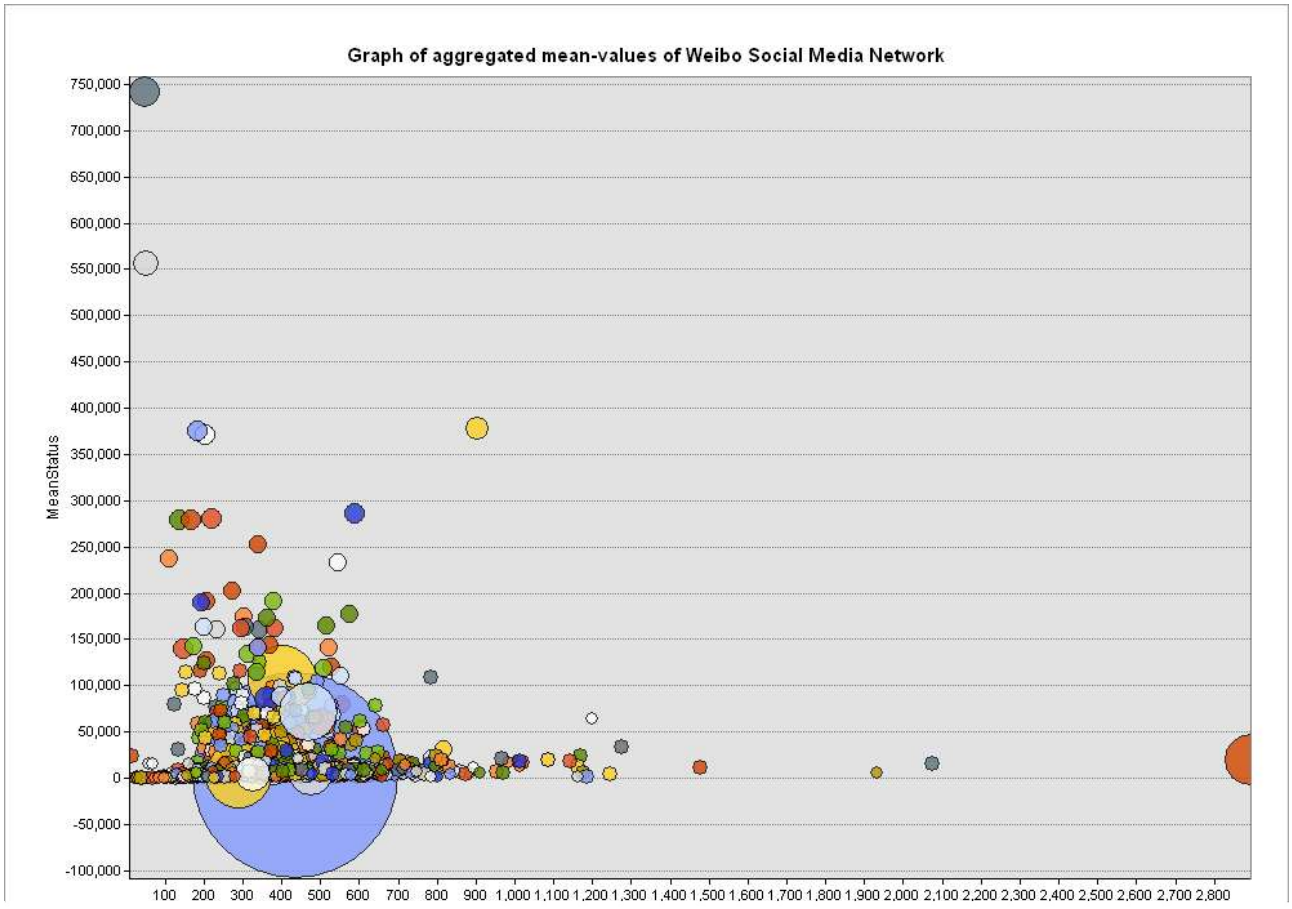
Appendix 1: Similar Weibo Users in spatial connected regions at smaller clusters (Group6 & 8)



Appendix 3: Group Distributions by count



Appendix 5: Plotting of aggregated Weibo Social Media Data - Meanfriend per Size



Appendix 4: Graph of aggregated mean-Values - MeanStatus per MeanFriend, Size = MeanFollower