applied sciences

MDPI

Article

Estimating Road Segments Using Natural Point Correspondences of GPS Trajectories

Artem Leichter [1,*] and Martin Werner [2]

[1] Institute of Cartography and Geoinformatics, Leibniz University Hannover, 30167 Hannover, Germany
[2] Institute for Applied Computer Science and Forschungsinstitut CODE, Bundeswehr University Munich, 85577 Munich, Germany; martin.werner@unibw.de
* Correspondence: artem.Leichter@ikg.uni-hannover.de

Received: 31 August 2019; Accepted: 8 October 2019; Published: 11 October 2019

Abstract: This work proposes a fast and straightforward method, called natural point correspondences (NaPoCo), for the extraction of road segment shapes from trajectories of vehicles. The algorithm can be expressed with 20 lines of code in Python and can be used as a baseline for further extensions or as a heuristic initialization for more complex algorithms. In this paper, we evaluate the performance of the proposed method. We show that (1) the order of the points in a trajectory can be used to cluster points among the trajectories for road segment shape extraction and (2) that preprocessing using polygonal approximation improves the results of the approach. Furthermore, we show based on "averaging GPS segments" competition results, that the algorithm despite its simplicity and low computational complexity achieves state-of-the-art performance on the challenge dataset, which is composed of data from several cities and countries.

Keywords: GPS; trajectory; segments; averaging; road network

1. Introduction

The automated extraction of road networks and their attributes is crucial for the success of challenging applications like dynamic routing, autonomous driving or artificial intelligence (AI)-driven urban planning. The pure size and rapid changes of the infrastructure make it impossible to survey road networks manually in required spatial resolution, level of detail, and update frequency. Data-driven approaches have been developed to tackle the automated generation or to update existing road networks. A typical strategy is to use remote sensing data [1,2] or to gather information from traffic participants in the form of trajectories which then can be processed to obtain road segments. Trajectories are defined as a list of waypoints in metric space. Sometimes the waypoints include further information such as the traveling speed or the type of traffic participant which can be included in the processing. There are several ways to obtain trajectories of traffic participants. One way is to use external sensors [3], like mounted cameras that track the location of traffic participants. Another way is to use data, captured by vehicles that log their position and possibly additional information [4]. Similarly, the position of pedestrians, bikers or car drivers is often logged by mobile devices tracking the positions of their users. Since commodity devices with localization functionality are nowadays widely spread, the resulting data becomes more and more popular for the task of generating road networks. Nevertheless, the quality of trajectories captured with mobile devices in urban regions is often not good enough to obtain precise information about the real position of the user in order to reconstruct the road network. On the one hand, this is due to low-cost sensors, on the other hand, the movement of traffic participants such as pedestrians usually does not correspond directly with the middle of the road. Furthermore, the data can be obfuscated to reduce privacy risks [5]. The idea to overcome these problems is to use multiple trajectories and predict the real position of the road. This task is called trajectory averaging.

Furthermore, since a lot of data is available, fast processing of the data using efficient algorithms is often as important as achieving accurate results. With these challenges in mind, the "averaging GPS segments" contest [6] was organized with the task of averaging segments of GPS trajectories to predict road segments while including the algorithms speed in the final evaluation. This work is mainly based on our submission of this contest.

Task

The process of automated extraction of road networks from trajectories can be divided into two main steps (1) estimation of intersections and (2) estimation of road segments between them. In this paper, we focus on the second part. Thus, the addressed task is defined as follows. We are given a set of trajectories which in the ideal case all belong to one road segment and each trajectory consisting of an ordered set of 2D points which represent the two-dimensional position on the lateral plane. The objective is to determine the shape of the road segment. Based on the assumption that large enough sets of gathered trajectories vary around the true location of the road segment, the objective can be reduced to determine a sensible summary C of the trajectory set $X = x_1, x_2, ..., x_k$. This can be achieved by solving the optimization problem:

$$\text{argmin}_C \sum_{i=1}^{k} D(x_i, C)^2$$

Due to the different number of points in trajectories and no missing information about the correspondences of points among the trajectories the solution for the optimization problem has to be searched in a huge configuration space resulting in at least quadratic run time for an optimal solution. We propose to assume correspondences of the points among the trajectories based on their order and then determine the shape of the road segments by the less complex averaging of the corresponding points. This simple approach reaches good results on the data provided for the contest, has a very good run time behavior and is conceptually simple in that it allows modifications and extensions in a straightforward manner. In this work, we evaluate the performance of our approach and compare it to the results of the averaging competition.

The remainder of the paper is organized as follows. Related work is presented in Section 2. The details of the method are described in Section 3. Experiments performed are described in Section 4 and conclusions are drawn in Section 5.

2. Related Work

For the aspects of road network extraction thematized in the contest, the most challenging problem is to determine correspondences across trajectories to be able to determine the points of the road segment polyline. It is not trivial to integrate the usage of the spatial information from trajectory points and their order within the trajectory. This fact leads to approaches that ignore the order of points in the trajectory [7], use available topology information [8,9], or apply regression [10]. The Medoid approach [11] focuses on the selection of one of the input trajectories without combining the information from several trajectories. More advanced strategies consider trajectories as time series and apply methods based on dynamic time warping (DTW) [12], discrete rasters like it is proposed in CellNet [11]. In [13] an iterative method is proposed with an adapted version of kernelized time elastic averaging (iTEKA) [14] In contrast to existing approaches our algorithm sticks to the order of the points in trajectories to cluster them for the averaging.

3. Method

The main idea of our method, called NaPoCo, is to calculate a simple average from subsets of the points from the trajectory set. We use the order of the points in trajectories to determine the subsets for averaging. The input data for this algorithm is created by slicing the original trajectories into

intermediate parts between the estimated intersection. This means that we are only analysing a small period of time in which the cars would behave similarly in terms of speed acceleration and moving shape due to traffic regulations and the shape of the road. The variation of sampling rates has only little influence on the number of points per trajectory. In the evaluated data set, most trajectories with deviating number of points are outlier where the vehicle was not moving on the assumed road segment (for example, Figures A1 and A2). Ignoring the outlier trajectories, we assume that the order of points correlates with their position along the segment. The application Douglas–Peucker (DP) [15] helps to reduce the influence of sampling rate, as the resulting number of points is dependent on the complexity of the underlying shape. The correspondences between the points in different trajectories are even more clear as the points in optimal case refer to the start point, endpoint, and curves.

Given a set of trajectories T, the algorithm generates the averaged segment based on the following steps in Figure 1:

1. Generalize single trajectories using the DP algorithm. DP takes a curve composed of segments defined by 2D points and creates a sketch by repeatedly adding the point with maximal orthogonal error to the sketch unless this error falls below a fixed threshold ϵ.
2. Determine the trajectory T_{max} that contains the most points. Reverse the order of the points in some trajectories to achieve identical orientation. The order of indices in a trajectory t_i is reversed if the distance between the endpoint of t_i and start point of t_{max} is smaller than the distance between the start point of t_i and the start point of t_{max}.
3. Align indices of the points in trajectories with a smaller number of points along T_{max} using linear interpolation. This step was part of the contribution to the contest, but as described in the Experiments and Discussion part this step is not always contributing to the performance improvement.
4. Calculate the centroid of all points with the same index by averaging coordinates.

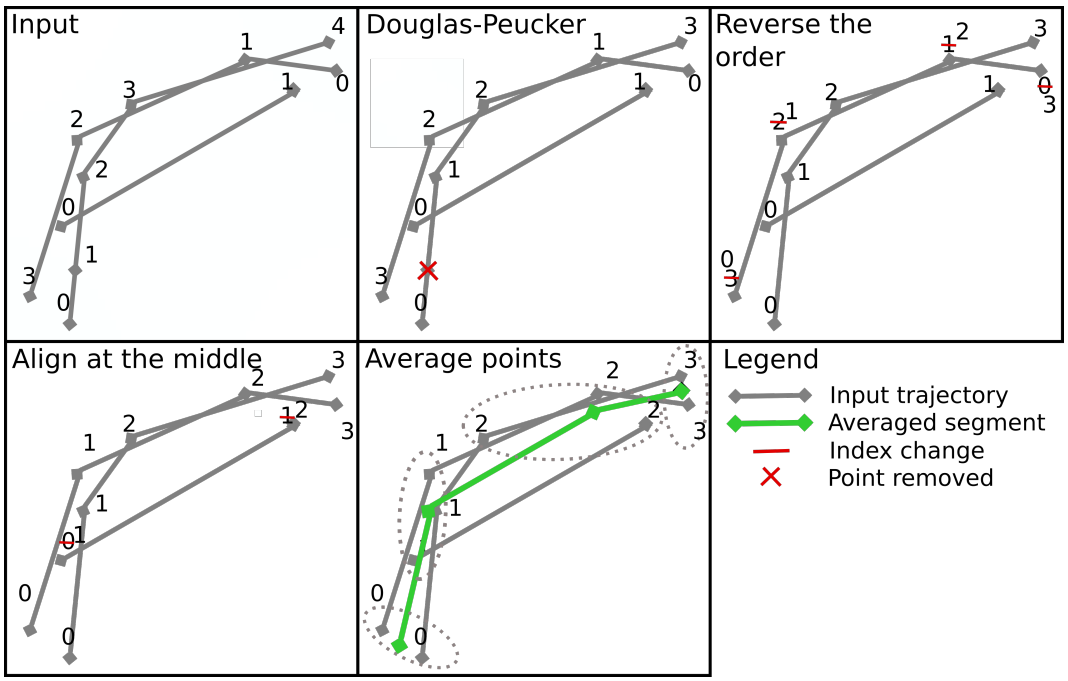

Figure 1. Visualization of the intermediate steps of the algorithm.

4. Experiments

For the experiments, we used the data set from the "averaging GPS segments" challenge (http://cs.uef.fi/sipu/segments/). This data set consists of samples from several open data sets (Joensuu, Chicago, Berlin, Athens) [6] collected in different countries. Examples from the data set are visualized

in the figures at the pages 8 and following. Each sample contained a set of trajectories mapped to a ground truth road segment which was extracted from OpenStreetMap. The coordinates of the points in the trajectories were normalized into the range from 0 to 1 and projected into a local Euclidean coordinate system. The quality of the predicted road segment was measured using the HC-SIM metric [6]. This metric, as an extension of the C-SIM [11] metric, uses a desensitization approach to determine the overlap of two rasterized and dilated lines, in this case, the predicted line and the reference line. The hierarchical part applies C-SIM with different grid resolutions and averages the determined similarity measure. In this work, we reported two sets of experiments. In the first experiment, we performed an ablation study which starts with the most simple version of the proposed algorithm and adds additional steps of the algorithm to determine the influence of each step on the performance of the algorithm. The second set of experiments was used to evaluate the influence of the ϵ parameter from the Douglas–Peuker algorithm. This value was varied in the range from 0.0 (no approximation) to 0.4. Since the coordinates are normalized, this value range is applicable to all samples regardless of the length of the segment in reality. In this experiment, we used the version of the algorithm without the alignment step as it turned out not to improve the results.

Both experiments were evaluated using the average HC-SIM measure over all 100 samples and by visually analyzing several results. Since the algorithm is deterministic, it was not necessary to evaluate the experiments multiple times.

4.1. Results

In the first experiment, the minimal version of the algorithm without DP, reversion of trajectory indices, and alignment reaches overall average HC-SIM of 0.53. Introducing the trajectory reversion step leads to an increased measure of 0.66. Finally, adding Douglas Peucker ($\epsilon = 0.15$) both with and without alignment of the indices provides the same HC-SIM value of 0.67 (Figure 2).

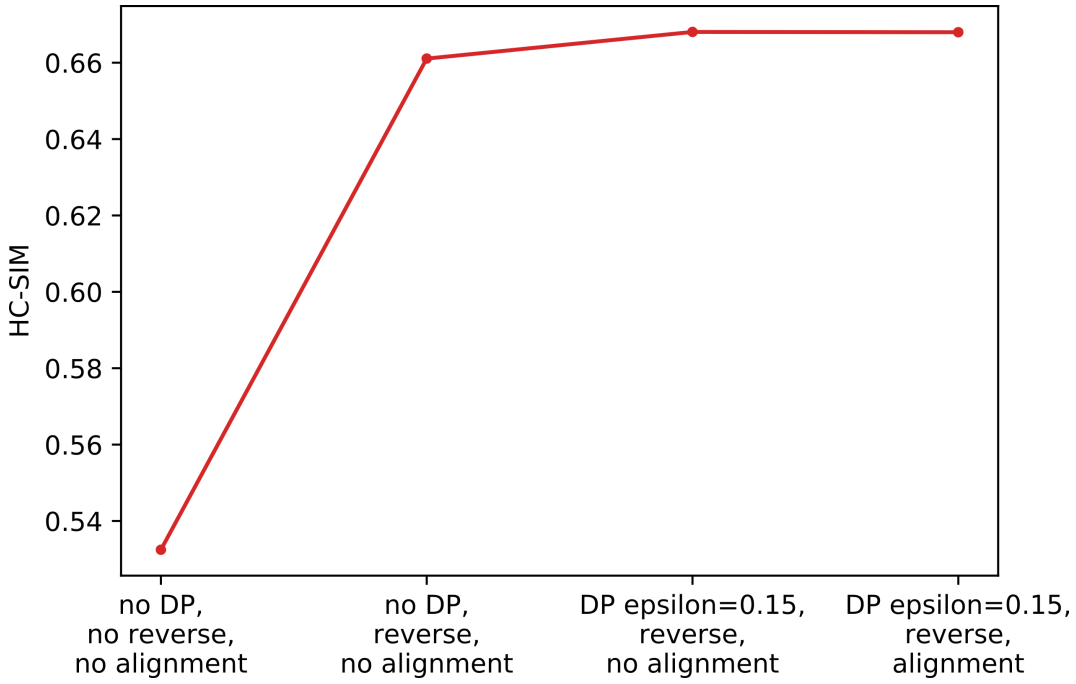

Figure 2. Average HC-SIM over whole data set for the algorithm after introducing additional steps.

The averaged similarity over the whole data set occludes drawbacks and improvements in particular constellations of samples. Therefore, we take a more detailed look at 5 selected samples (ids: 9, 14, 20, 49, 95, see Figure 3). Adding the reversion (step 2) of the input trajectories leads to an improvement in three out of the five exemplary cases. The results of the trajectories 20 and 95 (Figure A4) do not improve, because the input trajectories have all been observed in the same

direction. Among all 100 samples of the data set, there are no samples with a decreasing HC-SIM after the extension of the algorithm with reversion.

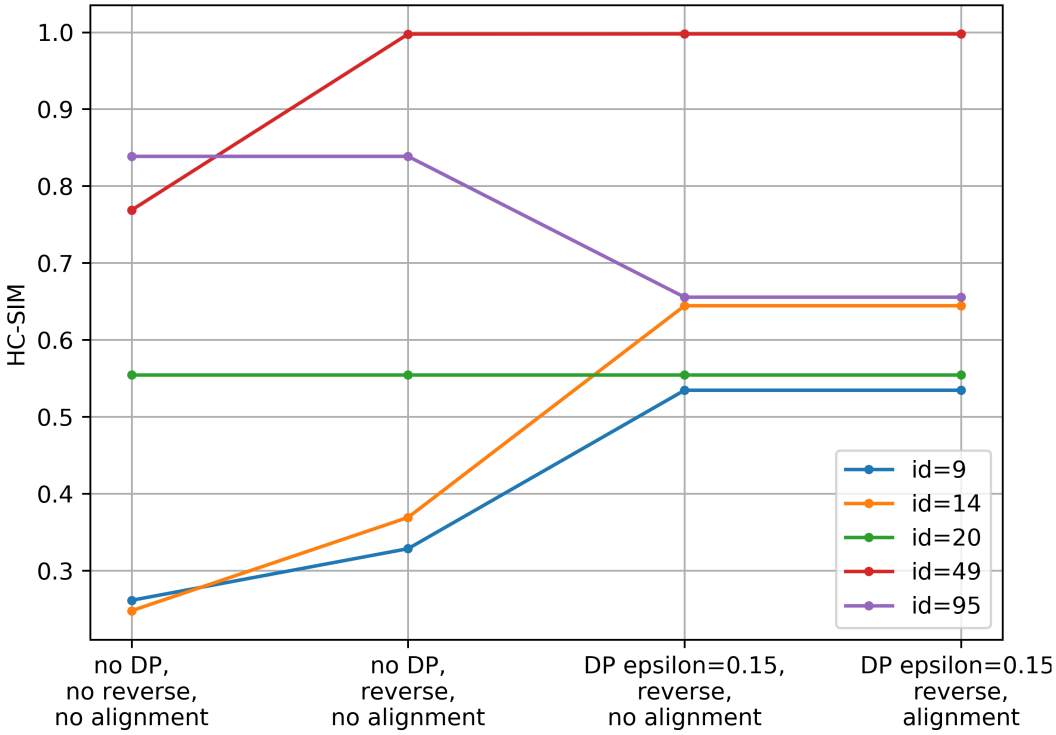

Figure 3. Similarity performance for five single samples.

The DP extension of the algorithm leads to an improvement for sample 9 (Figure A1) and sample 14 (Figure A2). While sample 14 improves due to the generalization of two input trajectories, sample 9 shows improved HC-SIM due to the simplification of the long input trajectory on the right side. This trajectory can be considered as an outlier and the reduction of the number of points of these trajectories by the DP reduced the influence of this outlier on the result. Samples 20 and 49 (Figure A3) were not influenced by applying DP simplification. In both cases, the reference segment, as well as the trajectories, were linear two-point segments. In that case, the approximation step has no influence. The DP has a negative impact on the segment with the id 95. This segment is U-shaped and reducing the number of input points makes the corners of the average drift away from the ground truth. The alignment of the trajectories has no influence on the result of any sample. For future work, we plan to allow segment models with constant curvature (e.g., polylines with bulges) as a generalization of line strings in order to simplify such curved road situations in a more sensible way.

The results of the second experiment were presented in two blocks as well. The first block shows the average HC-SIM over the entire data set and in the second block, the similarity values of five selected examples are shown. The HC-SIM behaves mostly predictable, with initially rising similarity values and after the ϵ of 0.15 reducing. There is still one outlier as the curve starts at 0.66 with no approximation and drops to a lower value near a value of $\epsilon = 0.04$ and starts to rise after that. The number of points after the approximation drops fast and converges to two (compare Figure 4).

The samples with ids 49 and 26 have constant HC-SIM values of approximately 1.0 and 0.28. The minimum at $\epsilon = 0.04$ is, however, present at the graph for the samples 10, 14 and 95. The similarity value of the samples with id 14 and especially 95 suffers from increasing ϵ (Figure 5).

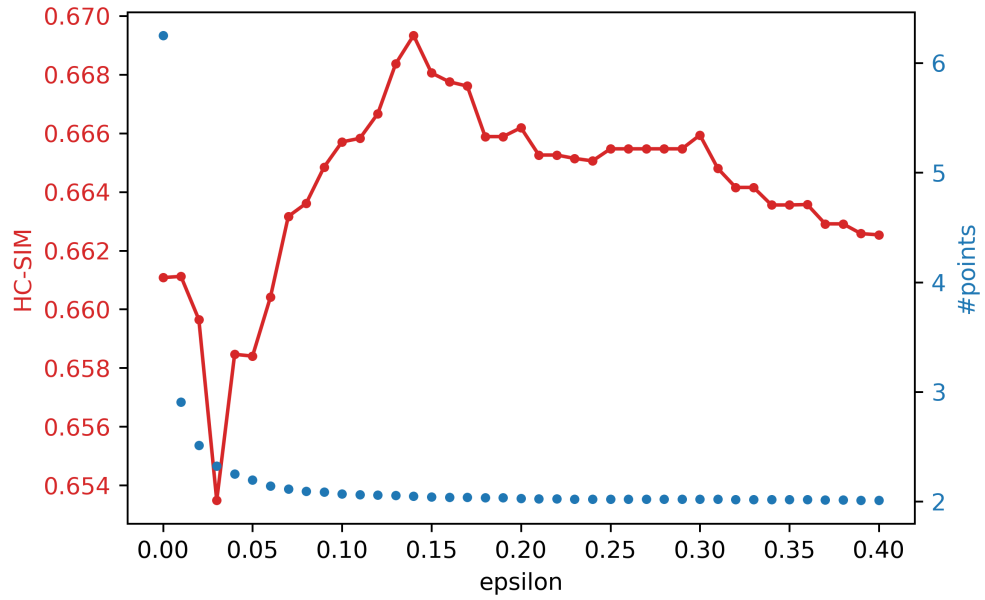

Figure 4. Average HC-SIM over the data set as function of Douglas–Peucker (DP) epsilon.

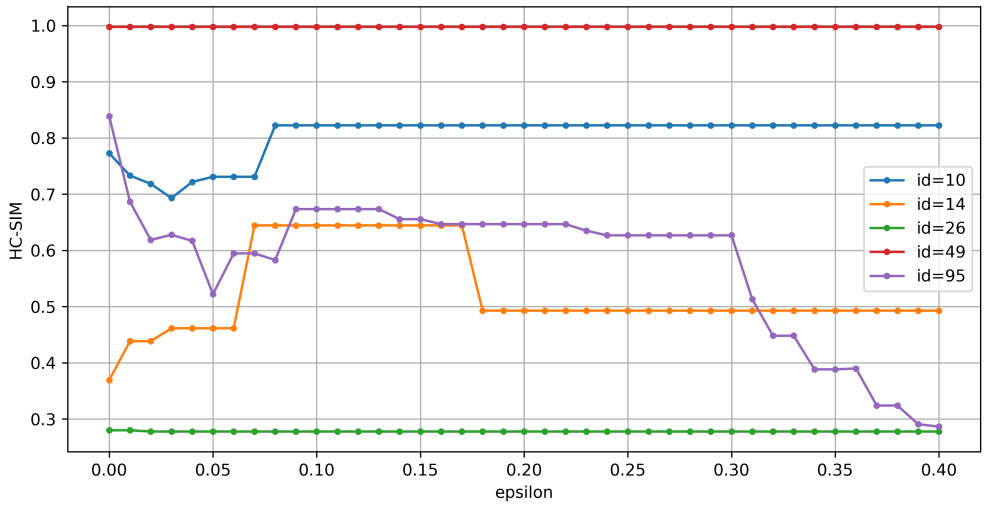

Figure 5. HC-SIM of single samples as function of DP ϵ.

4.2. Competition Results and Relative Performance

At the competition, this approach took the fifth place with only a little distance to the winner (See Table 1). The competition results are available at http://cs.uef.fi/sipu/segments/results.html [6]. For the competition ϵ was set to 0.1 and a HC-SIM of 0.666 was reached on the training set which we used for experiments presented in this work. On the test set of the competition, NaPoCo reached a score of 0.615. Commonly used approaches, like Medoid and CellNet, reached HC-SIM values of 0.571 and 0.647 respectively. The winning algorithm Korasek-2 reached a HC-SIM of 0.671 on the training set and 0.620 on the test set. This was only slightly higher then the performance of our method. It is worth noting that the top five submissions to this challenge were based on some variant of sequence averaging. Our method was the only one not using outlier rejection as well as subsampling the trajectory line segments.

Table 1. Competition results.

Method	Rank	Training	Testing	Runtime
Karasek-2	1	0.671	0.620	seconds
Yang-2	2	0.704	0.618	seconds
Yang-1	3	0.680	0.618	seconds
Leichter	4	0.666	0.615	seconds
Dupaquis	5	0.674	0.612	10 min
Amin	6	0.666	0.612	seconds
Karasek-1	7	0.681	0609	seconds
Medoid	–	0.619	0.567	1 h
CellNeto	–	0.664	0.612	seconds

4.3. Discussion

The reversion step of the algorithm is working well in all cases of the used data set. Nevertheless, in a situation where the end and the start of the trajectories are near to each other like e.g., in a deformed U-shape this approach could fail, because the distance between start and end of the real segment could be smaller then the variation of start and end point position due to noisy data. The alignment step is not necessary based on the results from this data set. There are no improved results after adding the alignment step to the basic algorithm (Figure A2). Most of the trajectories are reduced to two or three points after the approximation step and therefore, there is nothing to align. This is, however, a consequence of the very small road segments extracted for this challenge. In real-world scenarios, more complex shapes of roads might need to be formed. The results of the approximation step and its evaluation with varying ϵ values are not clear. Although using the Douglas–Peucker algorithm did improve the overall result (Figure A6) of the set, still many results suffer from the approximation (Figure A5). Especially the estimation of complex segments suffers from approximation with too high epsilon. On the other hand, at the moment, many samples suffer from the similarity break-in at $\epsilon = 0.04$. The approximation reduces fewer points on curvy outlier trajectories (Figure A3) and those dominate after that the averaging result.

5. Conclusions

The proposed method provides good competitive results combined with high simplicity and run-time efficiency. These properties make it well usable as deterministic baseline or heuristic for initialization of optimization approaches or a fall back method for approaches that differentiate by the complexity of the sample.

There is a lot of potential for improvement. There is a high dependency on the quality and nature of the input data. A presented outlier filter to exclude trajectories with a dramatically larger number of points should improve the results, as well as a per sample selected ϵ for DP. From a more general point of view, this paper shows that the ordering of points in a trajectory is a meaningful feature in practice, which provides useful information for the clustering of the points among the neighborhood trajectories. Therefore, it would be a rational step to combine the order of the points with metrical coordinates in three-dimensional space in order to cluster the points from different trajectories.

Author Contributions: A.L. designed the algorithm, performed the experiments, and wrote the paper. M.W. supported with ideas, discussions and suggestions regarding both the approach and the writing.

Funding: The publication of this article was funded by the Open Access fund of Leibniz Universität Hannover. This work was partially funded by the Federal Ministry of Education and Research, Germany (Bundesministerium für Bildung und Forschung, Förderkennzeichen 01IS17076). We gratefully acknowledge this support.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

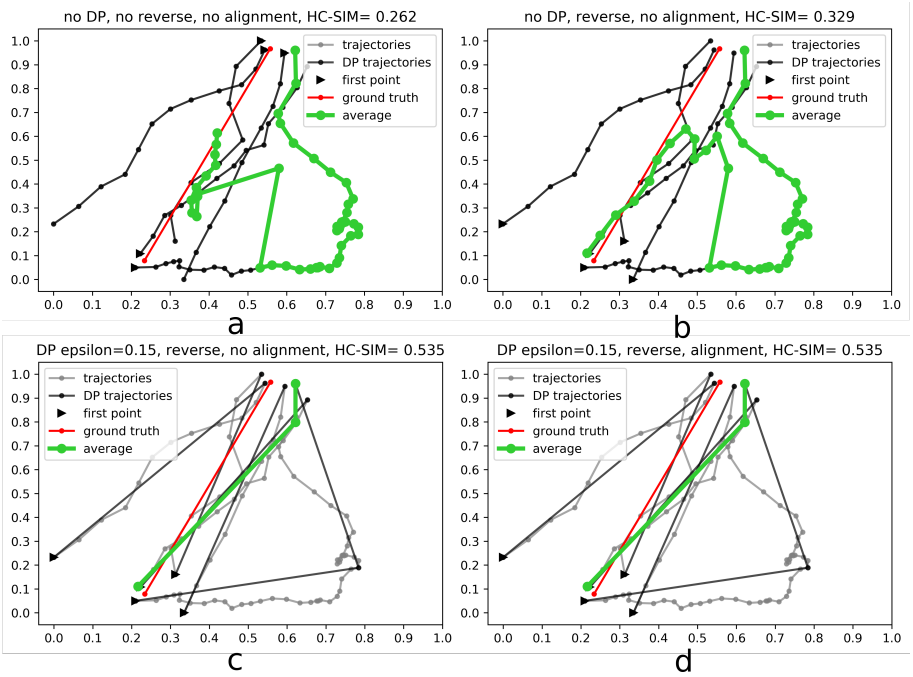

Figure A1. Algorithm applied to sample 9 with and without extensions. (**a**) No extensions just the averaging based on corresponding ids; (**b**) algorithm with reversion. The order of points in trajectory is reversed in order to group all starting points at one side; (**c**) algorithm with DP and ϵ set to 0.15; (**d**) algorithm including the alignment of the shorter paths.

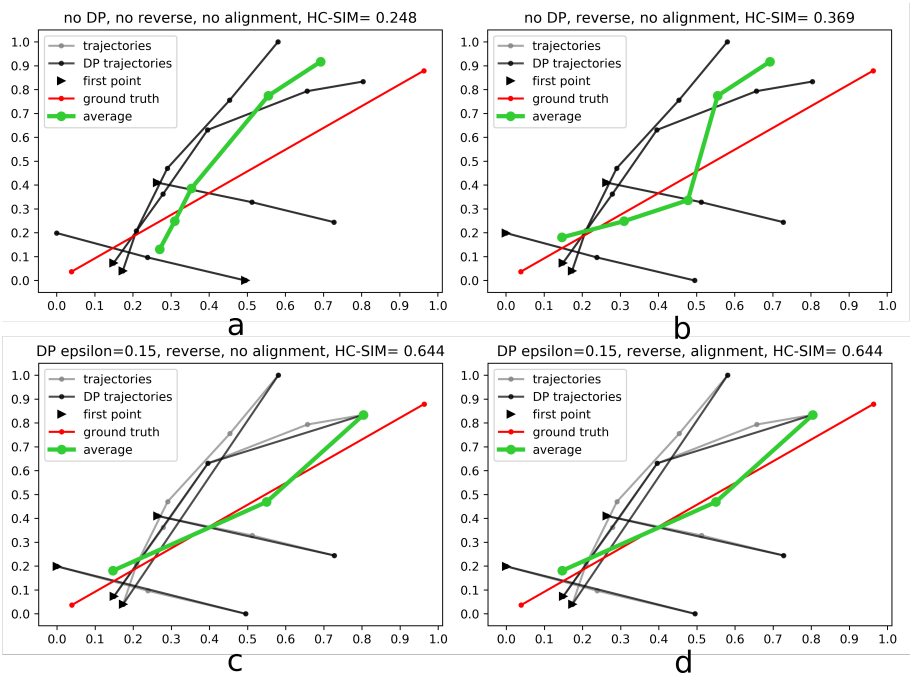

Figure A2. Algorithm applied to sample 14 with and without extensions. (**a**) No extensions just the averaging based on corresponding ids; (**b**) algorithm with reversion. The order of points in trajectory is reversed in order to group all starting points at one side; (**c**) algorithm with DP and ϵ set to 0.15; (**d**) algorithm including the alignment of the shorter paths.

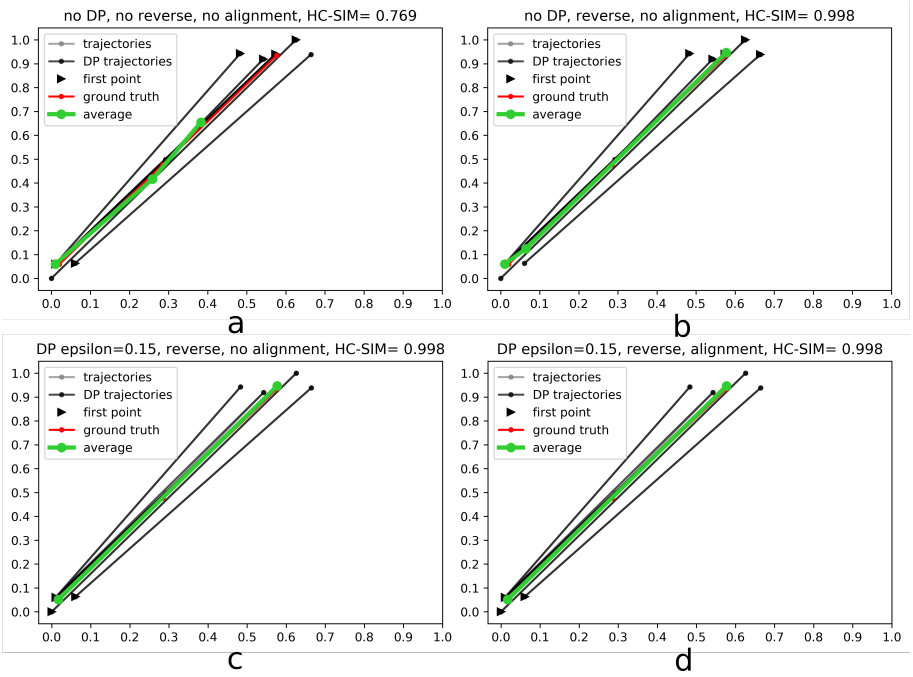

Figure A3. Algorithm applied to sample 49 with and without extensions. (**a**) No extensions just the averaging based on corresponding ids; (**b**) algorithm with reversion. The order of points in trajectory is reversed in order to group all starting points at one side; (**c**) algorithm with DP and ϵ set to 0.15; (**d**) algorithm including the alignment of the shorter paths.

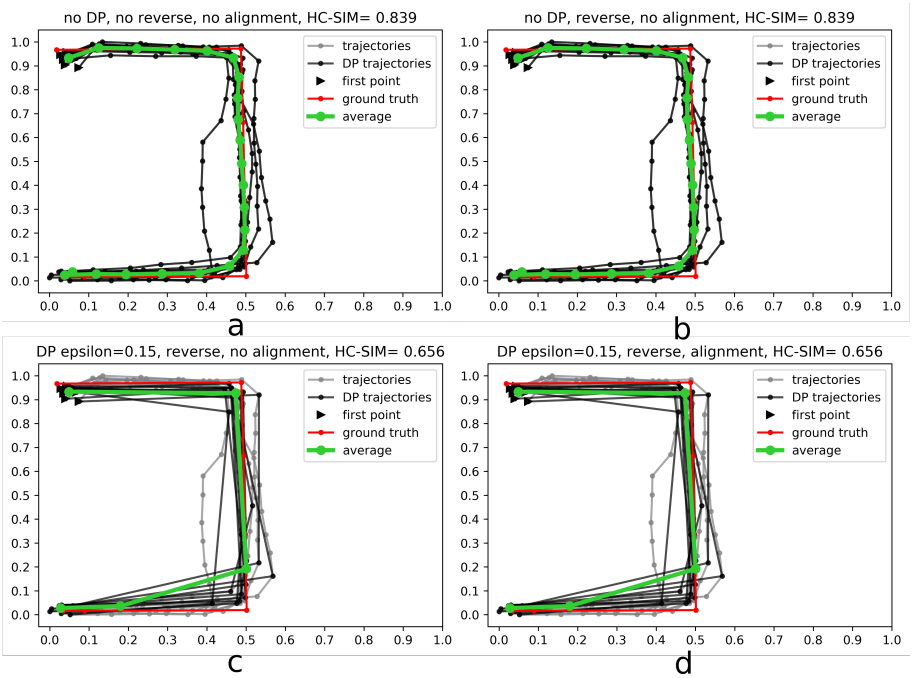

Figure A4. Algorithm applied to sample 95 with and without extensions. (**a**) No extensions just the averaging based on corresponding ids; (**b**) algorithm with reversion. The order of points in trajectory is reversed in order to group all starting points at one side; (**c**) algorithm with DP and ϵ set to 0.15; (**d**) algorithm including the alignment of the shorter paths.

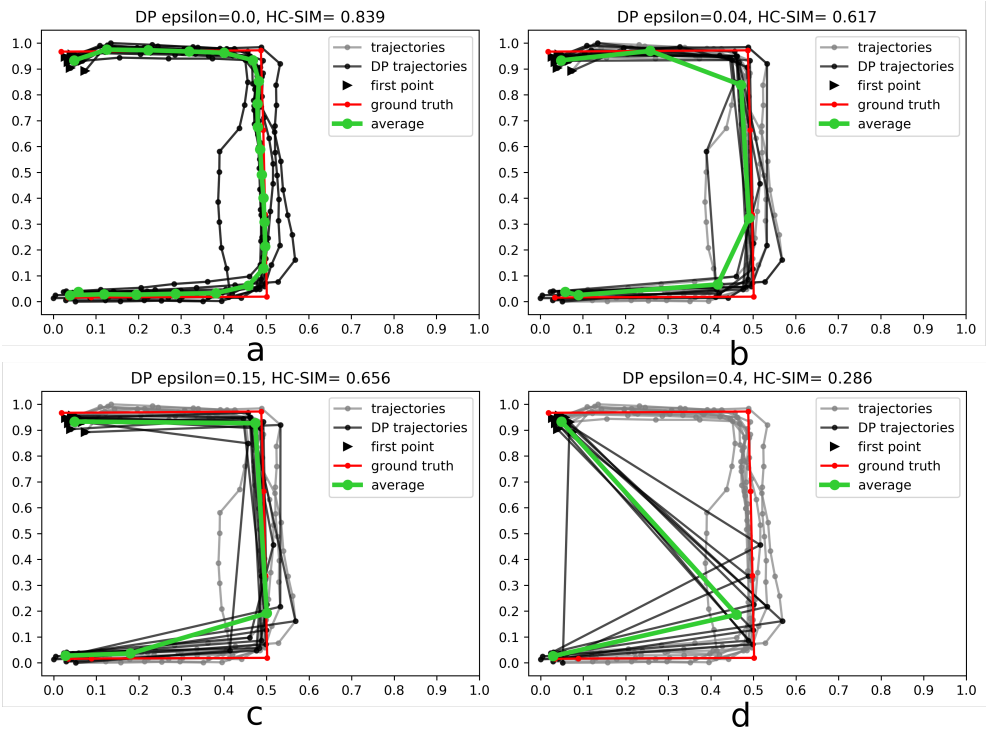

Figure A5. Algorithm applied to sample 95 with varying ϵ. (**a**) $\epsilon = 0.0$, (**b**) $\epsilon = 0.04$, (**c**) $\epsilon = 0.15$, (**d**) $\epsilon = 0.4$.

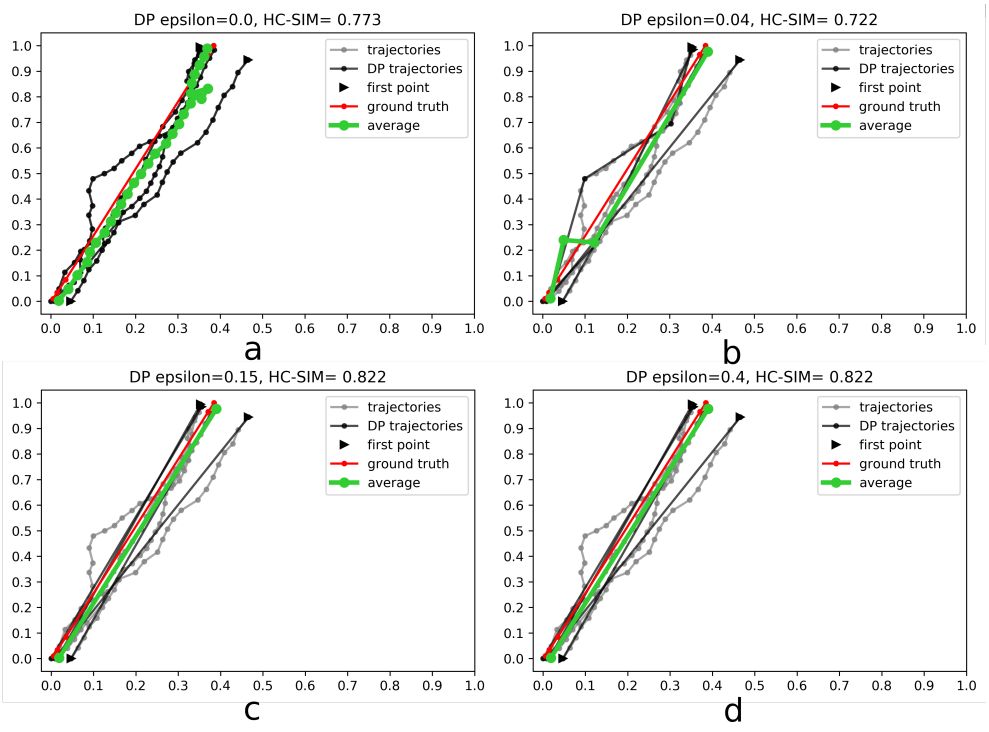

Figure A6. Algorithm applied to sample 10 with varying ϵ. (**a**) $\epsilon = 0.0$, (**b**) $\epsilon = 0.04$, (**c**) $\epsilon = 0.15$, (**d**) $\epsilon = 0.4$.

References

1. Zhang, Y.; Xiong, Z.; Zang, Y.; Wang, C.; Li, J.; Li, X. Topology-aware road network extraction via Multi-supervised Generative Adversarial Networks. *Remote Sens.* **2019**, *11*, 1017, doi:10.3390/rs11091017. [CrossRef]

2. Li, P.; Zang, Y.; Wang, C.; Li, J.; Cheng, M.; Luo, L.; Yu, Y. Road network extraction via deep learning and line integral convolution. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 1599–1602, doi:10.1109/IGARSS.2016.7729408. [CrossRef]
3. Koetsier, C.; Busch, S.; Sester, M. Trajectory extraction for analysis of unsafe driving behaviour. *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2019**, *XLII-2/W13*, 1573–1578, doi:10.5194/isprs-archives-XLII-2-W13-1573-2019. [CrossRef]
4. Zourlidou, S.; Sester, M. Intersection detection based on qualitative spatial reasoning on stopping point clusters. *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2016**, *XLI-B2*, 269–276, doi:10.5194/isprs-archives-XLI-B2-269-2016. [CrossRef]
5. Seidl, D.E.; Jankowski, P.; Tsou, M.H. Privacy and spatial pattern preservation in masked GPS trajectory data. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 785–800, doi:10.1080/13658816.2015.1101767. [CrossRef]
6. Fränti, P.; Mariescu-Istodor, R. Averaging GPS segments challenge 2019. Unpublished work, 2019.
7. Mariescu-Istodor, R.; Fränti, P. CellNet: Inferring Road Networks from GPS Trajectories. *ACM Trans. Spat. Algorithms Syst.* **2018**, *4*, 8:1–8:22, doi:10.1145/3234692. [CrossRef]
8. Ni, Z.; Xie, L.; Xie, T.; Shi, B.; Zheng, Y. Incremental road network generation based on vehicle trajectories. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 382, doi:10.3390/ijgi7100382. [CrossRef]
9. Li, D.; Li, J.; Li, J. Road Network Extraction from Low-Frequency Trajectories Based on a Road Structure-Aware Filter. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 374, doi:10.3390/ijgi8090374. [CrossRef]
10. Fathi, A.; Krumm, J. Detecting road intersections from GPS traces. In Proceedings of the International Conference on Geographic Information Science, Zurich, Switzerland, 14–17 September 2010; pp. 56–69, doi:10.1007/978-3-642-15300-6_5. [CrossRef]
11. Mariescu-Istodor, R.; Fränti, P. Grid-based method for GPS route analysis for retrieval. *ACM Trans. Spat. Algorithms Syst.* **2017**, *3*, 8:1–8:28, doi:10.1145/3125634. [CrossRef]
12. Velichko, V.M.; Zagoruyko, N.G. Automatic recognition of 200 words. *Int. J. Man-Mach. Stud.* **1970**, *2*, 223–234. [CrossRef]
13. Marteau, P.F. Estimating road segments using kernelized averaging of GPS trajectories. *Appl. Sci.* **2019**, *9*, 2736, doi:10.3390/app9132736. [CrossRef]
14. Marteau, P. Times series averaging and denoising from a probabilistic perspective on time-elastic kernels. *arXiv* **2016**, arXiv:1611.09194.
15. Douglas, D.; Peucker, T. Algorithm for the reduction of the number of points required to represent a line or its caricature. *Can. Cartogr.* **1973**, *10*, 112–122. [CrossRef]