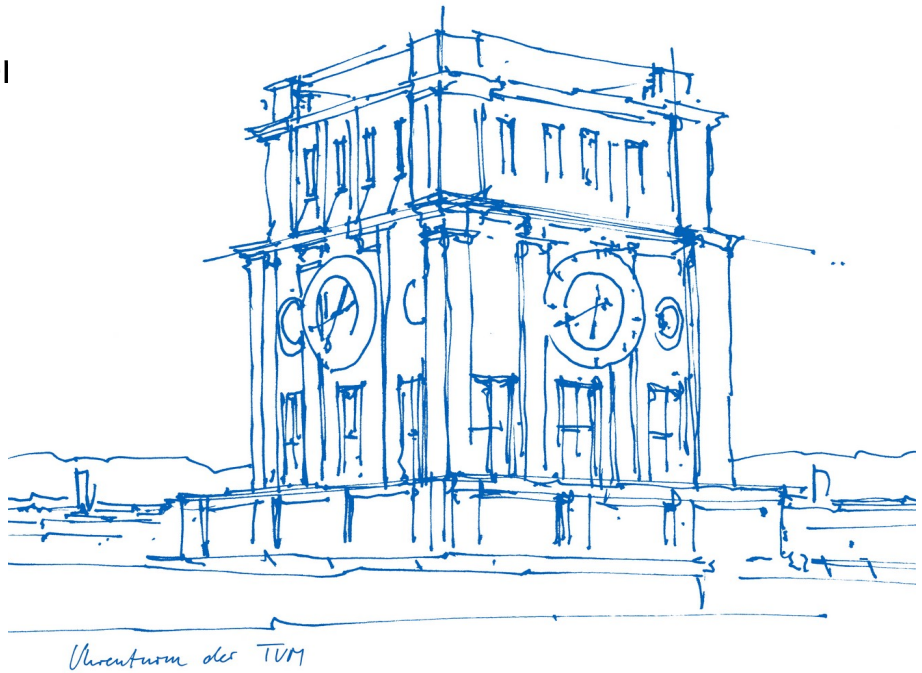
Improving persistence based trajectory simplification

Moritz Laass

Technische University of Munich

TUM Faculty of Aerospace and Geodesy

Professorship of Big Geospatial Data Manage



About Me

Moritz Laass
moritz.laass@tum.de

Since 2020: Scientific Assistant / soon to be PHD student
2014 - 2017: Msc. Human Computer Interaction LMU Munich

Overview

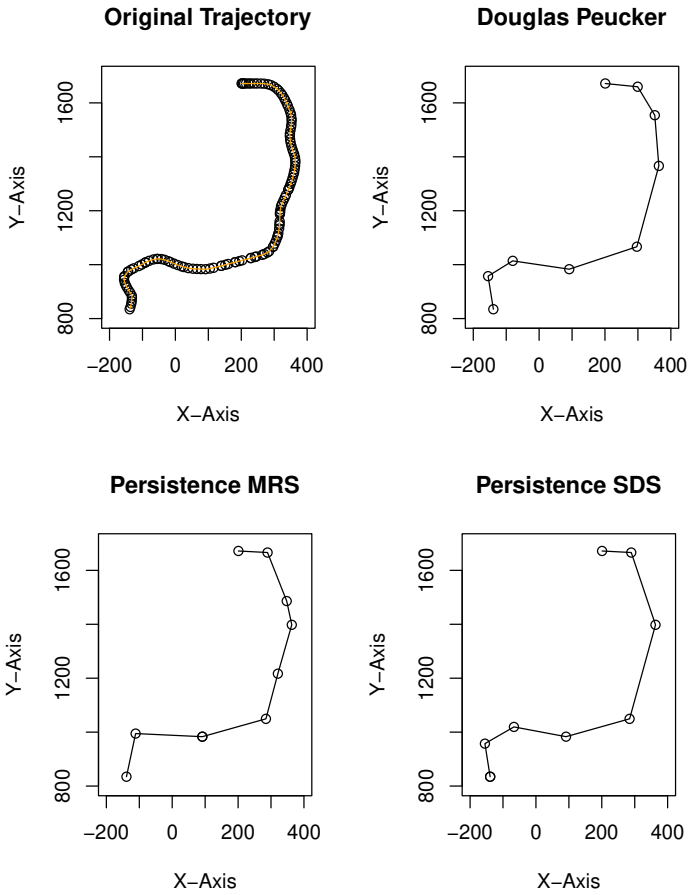
1) Introduction
   - Trajectory Definition
   - Trajectory Simplification
   - Persistence Algorithm
2) Improvements
   - Multi Resolution Simplification
   - Segment Distance Simplification
3) Evalutation
   - Datasets
   - Frechet Distance
   - Performance
   - Results
4) Conclusion

Plot of the San Francisco Dataset [6]

Introduction

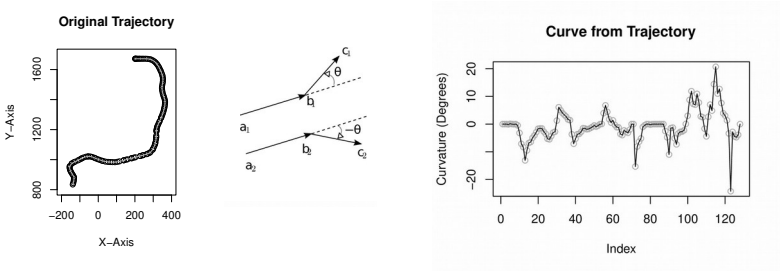
- Trajectory definition
  - Trajectory = Series of points in plane
  - Without timestamps
  - We assume interpolation between points is possible

- Simplification
  - Removing points from a trajectory while preserving the overall shape
  - Classical algorithm for this is Douglas Peucker which has a Complexity of $O(n^2)$

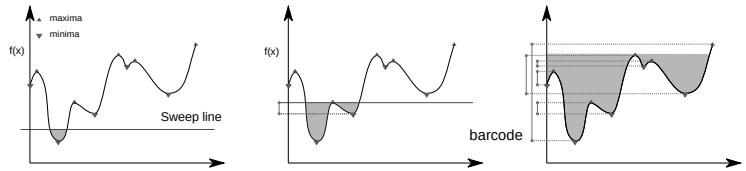


Persistence Algorithm for Trajectory Simplification

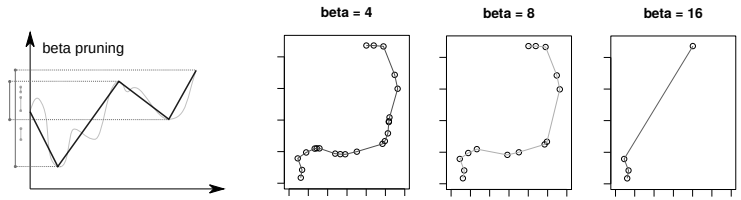
1) We **transform** a trajectory into 1D curve by taking the angle between neighbours in each point.



2) We apply the **persistence algorithm** [7], which calculates a set of connected components, the **barcode**, that connect each minimum with a maximum of the curve.



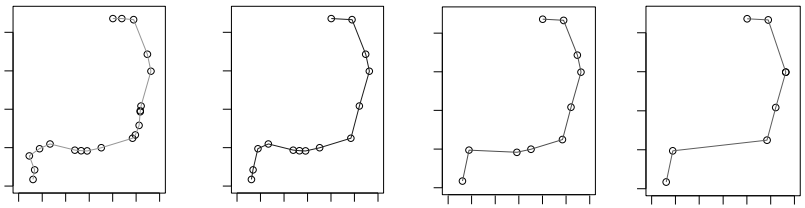
3) We perform **beta pruning** by removing all bars with a size smaller beta.



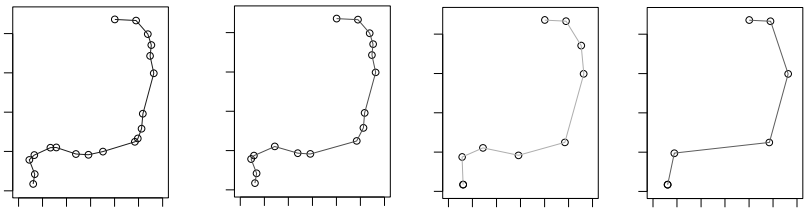
Persistence is a concept from the mathematical Field of topology, it has many applications. Here we use it to simplify a trajectory.

Improvements

Multi Resolution Simplification with different distances



Segment Distance Simplification with different distances



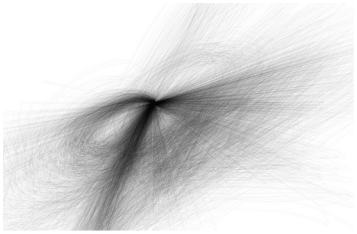
- One Problem of beta pruning is susceptability to noise.
- The original paper suggested **Multi Resolution Simplification (MRS)** [7] which uses the distance between points to prune noise artifacts which results in colinear points not being pruned.
- We introduce **Segment Distance Simplification (SDS)** which uses the Distance of a point to the line segment between the successor and predecessor points. Which alleviates the weakness of MRS.

Evaluation

We compared SDS performance to MRS and Douglas Peucker across a range of diverse Datasets using the frechet distance.

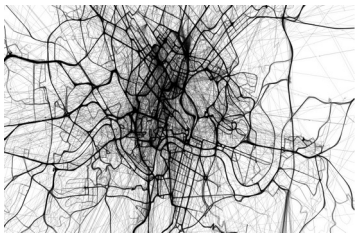
All trajectories where reduced to 100 points to compare the results.

Character	2,858 trajectories	handwriting	[1]
Geolife	25 mio. points	gps	[2]
Prague	250,000 points	ego-shooter	[3]
Roma	122 mio. points	gps	[4]
T-Drive	15 mio. points	gps	[5]
San Francisco	5 mio. points	synthetic	[6]



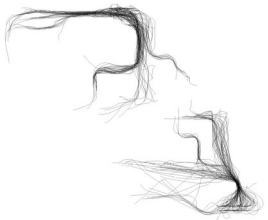
Character

Geolife



Roma

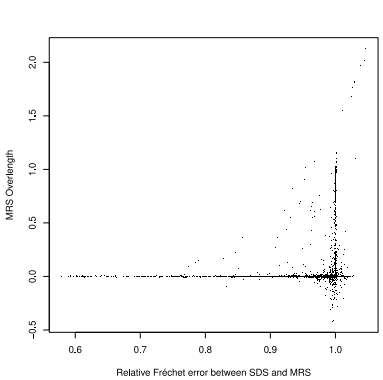
San Francisco

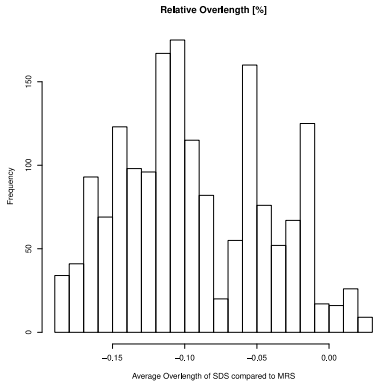


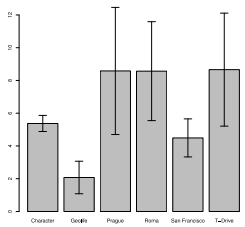
Prague

T-Drive

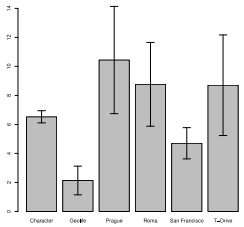
Results



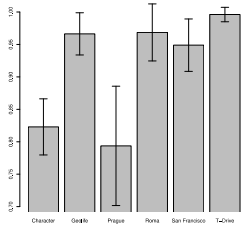




(d) SDS vs. DP



(e) MRS vs. DP



(f) MRS vs. SDS

- Figure 1 depicts the overall result distribution across all random parameters and all datasets comparing the relative performance gain of SDS compared to MRS
- For many cases of observed comparable error, MRS uses 5-15% more points.
- We compare the Fréchet distance of given algorithm pairs as the quotient
- MRS and SDS similar performance compared to DP (about 5-10 times higher error)
- SDS outperforms MRS significantly across all datasets with only very few examples where MRS found better simplifications

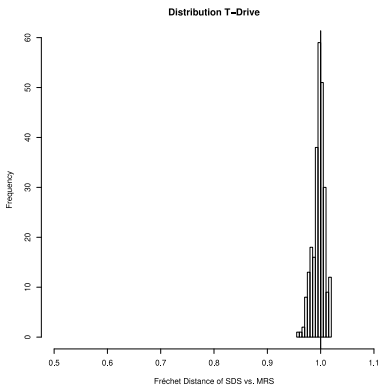
Perforamnce Distributions

Performance of MRS vs SDS using Frechet Distance as a relative metric

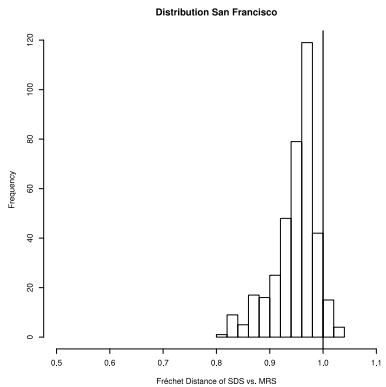
The vertical line at 1.0 signifies equal performance.

Notably both the character and Prague dataset SDS outperforms MRS in all cases.

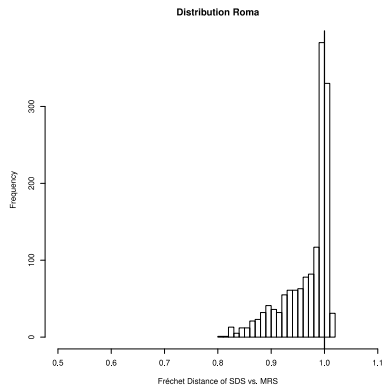
Both datasets are digitally acquired and have a high temoral resolution.



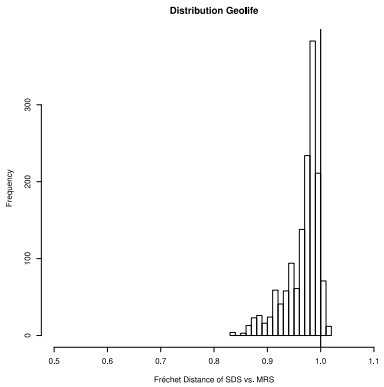
(a) T-Drive



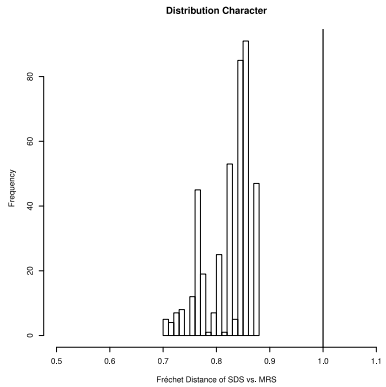
(b) San Francisco



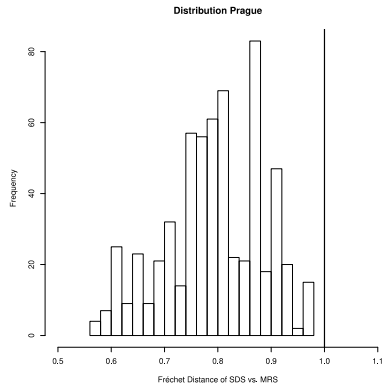
(c) Roma



(e) Geolife

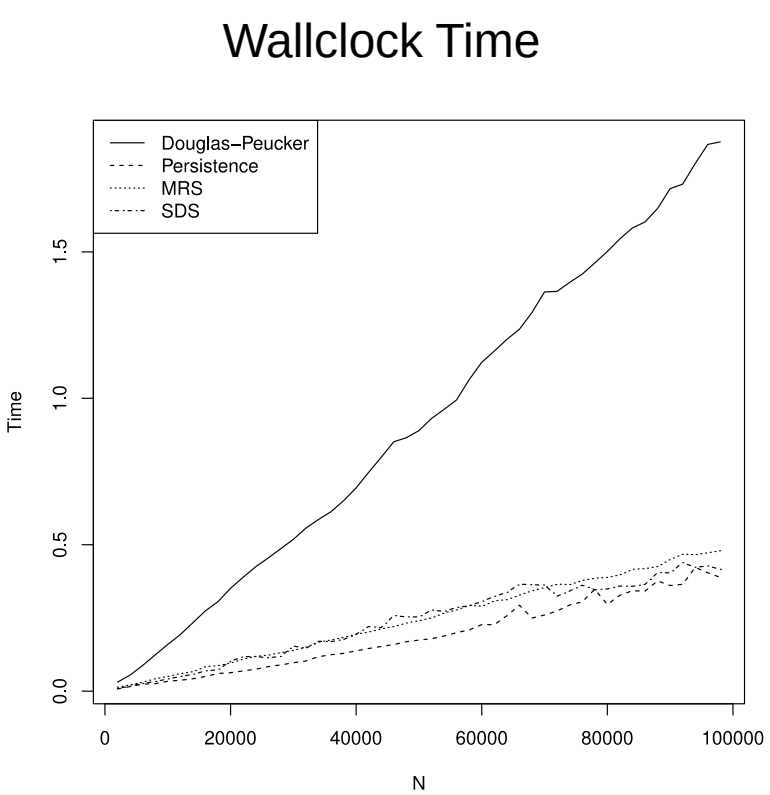


(f) Character

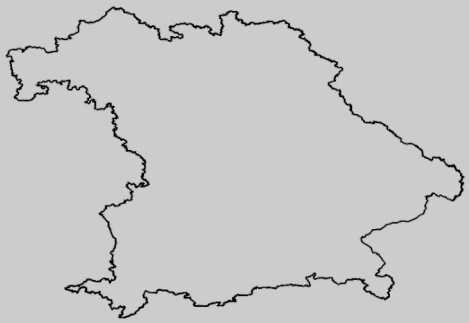


(g) Prague

Wallclock performance



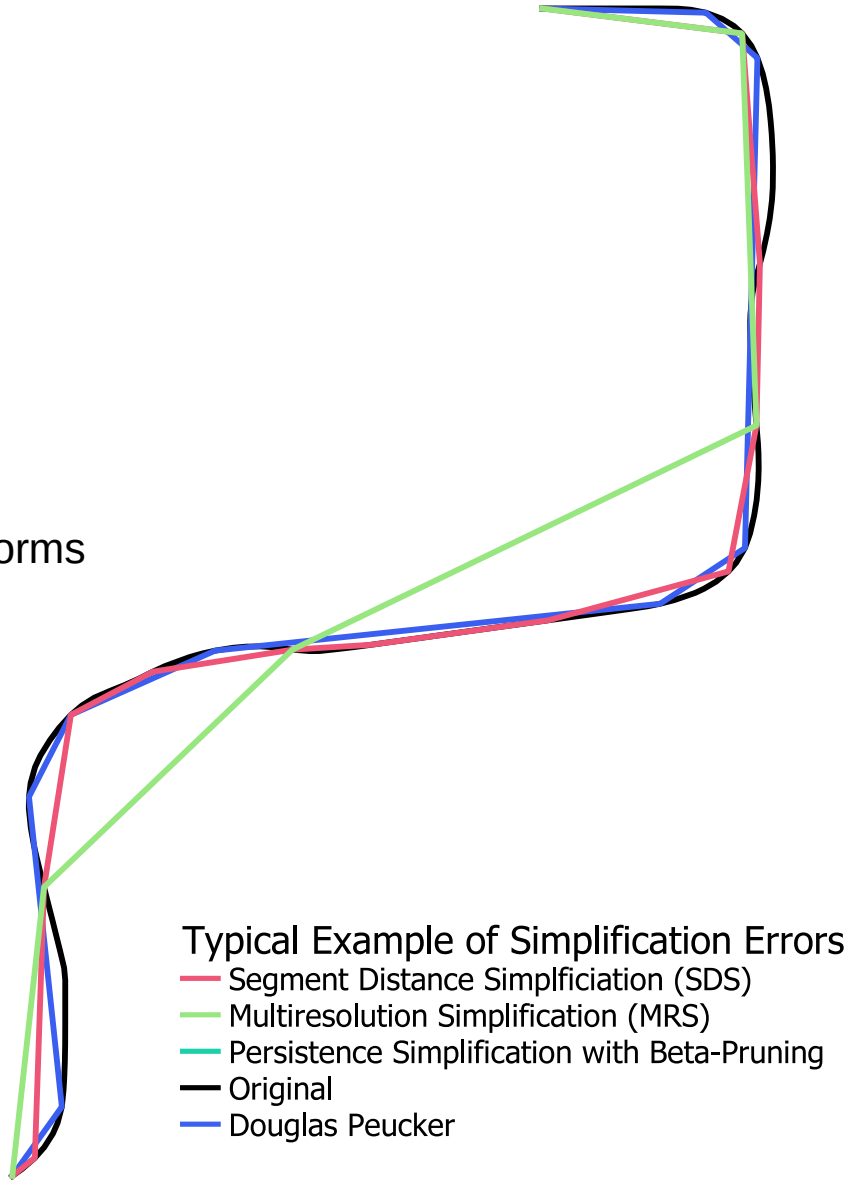
- Although we don‘t see the quadratic scaling of DP we can tell that the persistence algorithms significantly outperform DP
- Notably both MRS and SDS are very similar and only slightly slower than persistance without noise reduction



Bavaria boundary used for scalability analysis

Conclusion

- We confirm that Persistence based trajectory simplification outperforms Douglas Peucker in Computation time.
- Further we show that SDS significantly outperforms MRS in the majority of cases.
- We conclude that Persistence based trajectory Simplification using SDS can compete with Douglas Peucker, given the right parameter selection.



Thank you!

You can find furher information here
https://www.bgd.lrg.tum.de/resources/persistence.html

We also provide an open source implementation in C++
https://github.com/mlaass/persistence-open/

Bibliography:

[1] M. Lichman, “UCI machine learning repository,” http://archive.ics.uci.edu/ml, 2013.
[2] Y. Zheng, X. Xie, and W.-Y. Ma, “Geolife: A collaborative social networking service among user, location and trajectory.” IEEE Data Engineering Bulletin, vol. 33, no. 2, pp. 32–39, 2010.
[3] M. Werner, “Dataset containing trajectories of four players playing one hour of urban terror capture and hold on the map prague,” Available at http://trajectorycomputing.com/datasets/prague-ego-shooter-dataset/, 2014.
[4] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, “CRAWDAD dataset roma/taxi (v. 2014-07-17),” Downloaded from http://crawdad.org/roma/taxi/20140717, Jul. 2014. [5] J. Yuan, Y. Zheng, X. Xie, and G. Sun, “Driving with knowledge from the physical world,” in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011, pp. 316–324.
[6] M. Werner, “GISCUP 2015: Notes on Routing with Polygonal Constraints,” in SIGSPATIAL GIS CUP 15, in conjunction with 23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2015), 2015.
[7] P. Katsikouli, R. Sarkar, and J. Gao, “Persistence based online signal and trajectory simplification for mobile devices,” in Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2014, pp. 371–380.