Eurographics Conference on Visualization (EuroVis), Posters Track (2017)
A. Puig Puig and T. Isenberg (Editors)

Interactive Visualization of Massive Location Data using Multi-Scale Trajectories

T. De Bodt†[1,3], B. Adams[3] and J. Aerts[1,2]

[1]VDA-lab, ESAT/STADIUS, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven
[2]IMEC Smart Applications and Innovation Services, Kasteelpark Arenberg 10, 3001 Leuven
[3]Luciad, Arenberg Science Park, Gaston Geenslaan 11, 3001 Leuven

Abstract

Today location information on aircraft and vessels is collected worldwide and readily available to analysts. When analyzing such data for a large area or a long period of time, the sheer size of the dataset becomes a challenge. Especially when one wants to work interactively at both overview and detail scales. We present a scalable approach to visualize such data by treating it as a set of trajectories simplified at different error rates. When combined with tiling and GPU-based visualization, we can interactively and visually analyze a dataset of 1 billion position records on a workstation.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Hierarchy and geometric transformations

1. Introduction

A large amount of location data is collected every day. For example, more than 150000 ships worldwide broadcast their position continuously. Similarly, location data is available from airplanes, cars, persons, etc. Visualizing such data can provide valuable insights. In the field of visual analytics Shneiderman [Shn96] phrased the mantra "overview first, zoom and filter, details on demand". However, the sheer size of the data poses a significant challenge to answer the user's query in a timely fashion. An overview may require summarizing all data, but often allows lower accuracy, while a detailed view requires only a small part but at high precision. We propose a multi-scale representation for trajectory data by combining two well-known techniques: (1) multi-scale simplification and tiling for fast coarse-grained data filtering and aggregation and (2) GPU-based visualization for fine-grained interactive filtering.

2. Previous work

Image pyramids such as in Google Maps and Hotmap [Fis07] are widely used for efficient visualization. This concept of tiling and aggregation can also be applied without pre-rendering the tile content to a displayable image, which allows user-specific data filtering or styling. For example Immens [LJH13] uses binned aggregation to support interactive filtering of more than 100M point records. Aggregation can also be performed on-demand, but this requires a high performance system to keep the response time acceptable. For example the commercial MapD [RM16] system uses a GPU accelerated database to handle massive point datasets.

The line simplification algorithm by Douglas and Peucker [DP73] is widely known. However, it does not account for the time dimension in trajectory data. More recent algorithms often make use of the synchronized Euclidean distance (SED), which is based on distances between points at the same time instant. Muckel et al. [MHLR10] [MOJH*14] evaluated many algorithms and proposed an algorithm called SQUISH-E that uses a conservative error bound to remove points in a streaming fashion.

GPU based visualization of trajectory data is described by Buschmann [BTLD14]. By loading both the geometry and attribute data in GPU memory, styling and filtering rules can be evaluated in parallel for many data points. Scheepens [SWVdW*11] described a flexible framework for density-based visualization of trajectory data and showed how valuable insights can be gained.

3. Approach

We focus on trajectory data, a set of multivariate time series. Each entity reports its position at a number of time instants. Many existing systems can handle such data by simply considering it as a

† Corresponding author: thomas.debodt@kuleuven.be

DOI: 10.2312/eurp.20171170

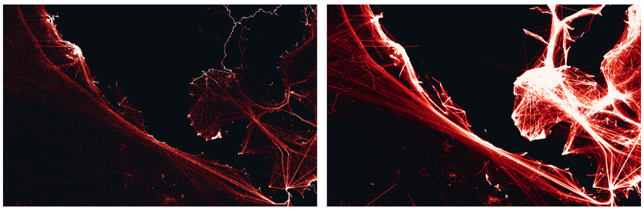

Figure 1: *Overview of maritime traffic for 1 month (1 km error bound) as a heat map: point-based (left) and line-based (right).*

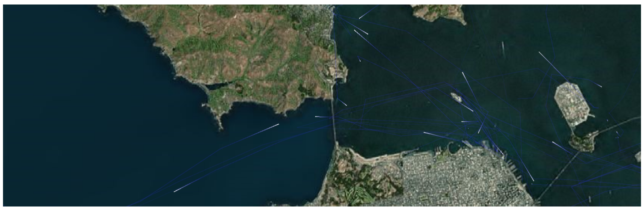

Figure 2: *Line-based visualization of 3 hours of maritime traffic near San Francisco (10 m error bound).*

set of individual points. However, this hides the connectivity between subsequent positions. One consequence is that a point-based heat map does not accurately show the amount of traffic if the report frequency is low or variable. For example, in vessel data this happens at open sea where the location is only recorded when a satellite passes by.

The trajectories are preprocessed to a multi-scale and tiled format. We selected SQUISH-E [MOJH*14] to simplify the trajectories at increasing error rates. This algorithm is simple, fast ($\Theta(NlogN)$) and provides a guaranteed error bound. In our algorithm, scales and their maximum error are chosen such that the coarsest yields sub-pixel accuracy when the entire dataset is shown, the finest corresponds to the original data and we have roughly 50% data reduction rate between 2 scales. Resulting multi-scale trajectories are then split into regularly sized tiles along the temporal and positional axes. We only store position (x,y) and time (t) for each point. Other attributes such as ship type and destination are stored separately since they change infrequently.

Similar to [CXF12] we apply a Mercator projection before simplification. In this projection, a path with constant bearing is a straight line. However, distance varies in a nonlinear fashion. For performance reasons we use a piecewise approximation with error $< 0.5\%$.

To visualize, we first select a scale such that $error_{scale} < \sqrt{Area_{pixel}}$. This ensures a deviation $< 1px$ for each line segment. For example on an HD screen showing the whole world 1 pixel covers $20km \times 20km$, so a simplification error $< 20km$ is accepted. Next, we select the tiles overlapping with both the spatial extent of the map and the selected temporal range. Finally, we load each tile in GPU memory (unless cached) and render using OpenGL line primitives.

4. Evaluation

We evaluated the approach with a maritime dataset provided by Spire. Our test system has an Intel Core i7-6700HQ and NVIDIA GeForce GTX 1080. The dataset consists of all AIS (Automatic Identification System) messages received in October 2016 by satellites (worldwide) and radars (US west coast only): 66k ships and 42M position reports. Preprocessing this data using our approach on 1 thread takes 51s and 49s for simplification and tiling respectively. Table 1 shows the results and figure 1 shows an overview heat map indicating the amount of traffic over the entire month. You can clearly see the effect of sample frequency on the point-based heat map. Figure 2 shows a detailed view. In general $< 64MB$ tiled data are needed (in memory) for visualization on an HD screen at any location or scale.

error bound	#records	compression	avg error	size
original data	42.4 M	-	-	899 MB
10 m	19.2 M	1/2	6 m	328 MB
100 m	6.9 M	1/6	55 m	115 MB
1 000 m	3.0 M	1/14	509 m	48 MB
10 000 m	1.3 M	1/33	4866 m	22 MB

Table 1: *Results for our maritime dataset: for each scale the SQUISH-E error bound, number of position records, compression ratio, actual average trajectory error and data size.*

To test scalability we increased the dataset to 2 years by replicating it for each month. The simplification algorithm only uses information in a small time window, so the same compression characteristics apply. The result is a dataset 24x larger: 1B position reports. For a world scale visualization with our approach, we load 528 MB of tiled data which is rendered at ± 30 FPS.

5. Conclusions and future work

We have shown that visualization of trajectory datasets of up to 1B locations on a spatial map is feasible on a single workstation. The multi-scale and tiled representation allows to efficiently select a spatio-temporal subset of the data at the appropriate precision, avoiding the need to render or even load the entire dataset. The SQUISH-E simplification algorithm provides a guaranteed error bound, so the visualization remains accurate.

We believe that such a multi-scale approach also has potential for accelerating analysis of large trajectory datasets. For example [JYZ*08] uses trajectory simplification and a filter-and-refinement paradigm to efficiently find spatio-temporal clusters. Alternatively, one could perform the analysis based on the simplified data and accept a certain error. This would require support for error margins in the algorithm, such that the error of its result can be determined.

6. Acknowledgements

TDB is financed through the VLAIO Baekeland mandate VISAGE (HBC.2016.0592). We would like to thank Spire for providing the maritime dataset. The prototype makes use of LuciadLightspeed geospatial visualization components.

References

[BTLD14] BUSCHMANN S., TRAPP M., LÜHNE P., DÖLLNER J.: Hardware-accelerated attribute mapping for interactive visualization of complex 3d trajectories. In *Information Visualization Theory and Applications (IVAPP), 2014 International Conference on* (2014), IEEE, pp. 356–363. 1

[CXF12] CHEN M., XU M., FRANTI P.: Compression of gps trajectories. In *Data Compression Conference (DCC), 2012* (2012), IEEE, pp. 62–71. 2

[DP73] DOUGLAS D. H., PEUCKER T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization 10*, 2 (1973), 112–122. 1

[Fis07] FISHER D.: Hotmap: Looking at geographic attention. *IEEE transactions on visualization and computer graphics 13*, 6 (2007), 1184–1191. 1

[JYZ*08] JEUNG H., YIU M. L., ZHOU X., JENSEN C. S., SHEN H. T.: Discovery of convoys in trajectory databases. *Proceedings of the VLDB Endowment 1*, 1 (2008), 1068–1080. 2

[LJH13] LIU Z., JIANG B., HEER J.: immens: Real-time visual querying of big data. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 421–430. 1

[MHLR10] MUCKELL J., HWANG J.-H., LAWSON C. T., RAVI S.: Algorithms for compressing gps trajectory data: an empirical evaluation. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2010), ACM, pp. 402–405. 1

[MOJH*14] MUCKELL J., OLSEN JR P. W., HWANG J.-H., LAWSON C. T., RAVI S.: Compression of trajectory data: a comprehensive evaluation and new approach. *GeoInformatica 18*, 3 (2014), 435–460. 1, 2

[RM16] ROOT C., MOSTAK T.: Mapd: a gpu-powered big data analytics and visualization platform. In *ACM SIGGRAPH 2016 Talks* (2016), ACM, p. 73. 1

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on* (1996), IEEE, pp. 336–343. 1

[SWVdW*11] SCHEEPENS R., WILLEMS N., VAN DE WETERING H., ANDRIENKO G., ANDRIENKO N., VAN WIJK J. J.: Composite density maps for multivariate trajectories. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2518–2527. 1