The Journal of Open Source Software

fastgeotoolkit: A High-Performance Geospatial Analysis Library and Novel Route Density Mapping Implementation

Alexander Akira Weimer[1] and **Justin Abraham**[1]

1 University of Minnesota

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 19 August 2025
Published: unpublished

Summary

fastgeotoolkit is a high-performance JavaScript library that introduces a novel segment-based approach to GPS trajectory analysis and route density visualization.

This library addresses several key limitations of existing approaches to route density visualization, leveraging a segment-based algorithm that processes GPS trajectories as sequences of connected line segments. By analyzing route overlap at the segment level, fastgeotoolkit produces density visualizations that better reflect route usage patterns.

The implementation is able to run on the web or locally, and handles heavy datasets well. As a result, it is a tool accessible for both research and consumer-facing development applications.

Statement of Need

GPS route density visualization is essential for transportation research (Chan et al., 2025), urban planning (Feng et al., 2020), trail management (Shi et al., 2023), and movement ecology (Seidel et al., 2018). Accurate route usage quantification supports evidence-based infrastructure decisions and mobility behavior analysis.

Research Gap in Route-Based Analysis

Current geospatial analysis treats GPS trajectories as point collections, applying kernel density estimation or clustering algorithms (Xu et al., 2024). This introduces several problems when dealing with data where the linear route is the primary feature.

Sampling bias amplification: GPS sampling rates vary by device settings and signal conditions (Müller et al., 2022). Point-based methods make these inconsistencies worse, preventing accurate route comparison across datasets (Xu et al., 2024).

Parameter sensitivity: Results change dramatically based on kernel bandwidth and grid size choices (Thierry et al., 2013), making it difficult to establish consistent analysis methods across studies.

Software Ecosystem Limitations

Inadequate existing tools: Popular GIS software (QGIS (QGIS Contributors, 2022), R spatial packages (Pebesma, 2018; Pebesma & Bivand, 2005), Python scipy (Virtanen et al., 2020)) focus on static point analysis, not linear route patterns.

Computational barriers: Most trajectory tools require preprocessing or server infrastructure (Chan et al., 2025), limiting research accessibility.

Proprietary algorithms: Commercial platforms like Strava use linear route processing (Zhang et al., 2023) but keep implementations closed, creating gaps in open science.

Research Contribution

fastgeotoolkit provides the first open-source, segment-based route density algorithm for GPS track data, enabling: (1) sampling-rate-independent frequency estimation, (2) browser-native analysis workflows, and (3) standardized trajectory analysis methodologies.

Implementation

fastgeotoolkit addresses issues with existing heatmap implementations by treating GPS tracks as sequences of connected segments rather than point clouds. This approach provides more accurate route frequency analysis, and fastgeotoolkit implements it in such a way that it enables processing millions of tracks without preprocessing or server-side infrastructure.

Segment-Based Algorithm

fastgeotoolkit's core algorithm processes GPS tracks in three steps:

Track segmentation: GPS tracks are split into consecutive coordinate pairs representing individual route segments. Each segment connects two adjacent GPS points, preserving the linear structure of the original path.

Coordinate normalization: To handle GPS measurement noise, coordinates are snapped to a tolerance grid. This reduces minor variations from GPS accuracy limitations while maintaining route integrity with high fidelity.

Frequency calculation: Each segment is converted to a normalized string key for efficient storage and lookup. A hash map tracks how many times each unique segment appears across all input tracks. Each track's final frequency is the average frequency of its constituent segments.

This approach ensures route popularity reflects actual overlapping usage rather than GPS sampling artifacts. Routes that share the same path segments will have higher frequencies, while unique routes will have lower frequencies.

Performance and Architecture

The algorithm runs in $O(n \times m)$ time where n is the number of tracks and m is the average track length. Hash map lookups provide $O(1)$ average-case performance for frequency queries.

The core implementation is written in Rust for memory safety and performance, then compiled to WebAssembly using wasm-pack. This enables browser-native execution without server dependencies while maintaining near-native computational speed (Jung et al., 2023; *WebAssembly Core Specification*, 2019).

The Journal of Open Source Software

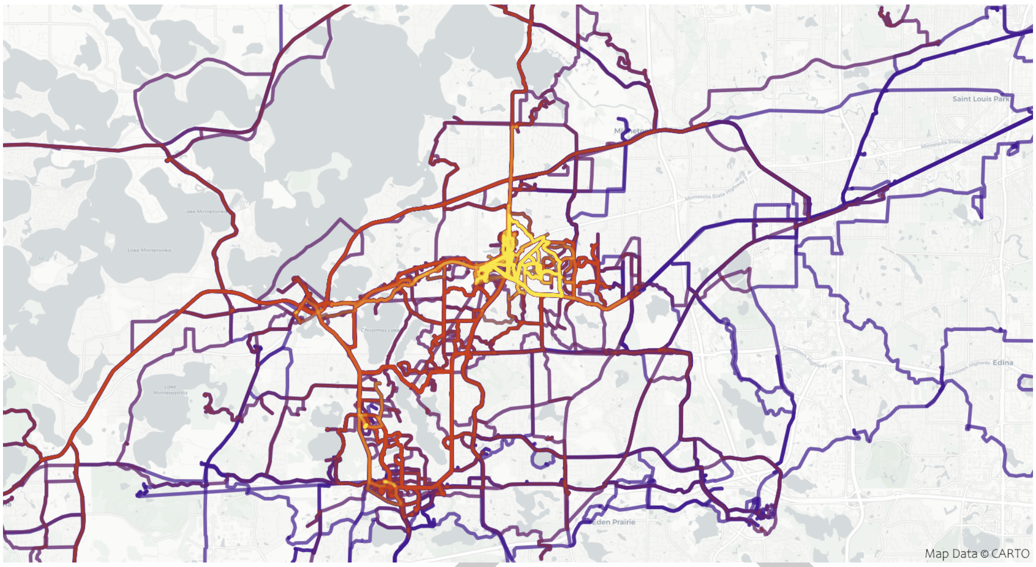

Figure 1: Example heatmap produced using fastgeotoolkit and MapLibre GL JS.

The library is distributed as an npm package[1] with TypeScript definitions, integrating naturally with existing JavaScript mapping libraries like Leaflet and MapLibre GL JS (*leaflet*, 2025; *maplibre-gl-js*, 2025). This allows for easy use in webapps as seen in Figure 1, a screenshot from the demo page for fastgeotoolkit.

Conclusion

fastgeotoolkit provides a practical solution for GPS route analysis by focusing on segments rather than points. This approach produces more accurate route density visualizations while being accessible through standard JavaScript tooling.

The segment-based algorithm handles the inherent challenges of GPS data, especially measurement noise, variable sampling rates, and device differences, without requiring complex preprocessing. fastgeotoolkit implements this approach while remaining highly performant, which makes it largely unique in the landscape of GIS tooling for the web.

Acknowledgements

The authors acknowledge the open-source geospatial community and the help of users who provided feedback during development.

References

Chan, T. N., Zhu, B., Wu, D., Peng, Y., U, L. H., Tu, W., & Wang, R. (2025). A Fast Line Density Visualization Plugin for Geographic Information Systems. In *ACM Conferences* (pp. 63–66). Association for Computing Machinery. https://doi.org/10.1145/3722212.3725075

Feng, Z., Li, H., Zeng, W., Yang, S.-H., & Qu, H. (2020). Topology Density Map for Urban Data Visualization and Analysis. *arXiv*. https://doi.org/10.1109/TVCG.2020.3030469

[1] Available at https://www.npmjs.com/package/fastgeotoolkit

Jung, R., Jourdan, J., Krebbers, R., & Dreyer, D. (2023). Safe Systems Programming in Rust. *Communications of the ACM*. https://cacm.acm.org/research/safe-systems-programming-in-rust

leaflet. (2025). https://github.com/Leaflet/Leaflet

maplibre-gl-js. (2025). https://github.com/maplibre/maplibre-gl-js

Müller, S. R., Bayer, J. B., Ross, M. Q., Mount, J., Stachl, C., Harari, G. M., Chang, Y.-J., & Le, H. T. K. (2022). Analyzing GPS Data for Psychological Research: A Tutorial. *Advances in Methods and Practices in Psychological Science*, *5*(2), 25152459221082680. https://doi.org/10.1177/25152459221082680

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *R Journal*, *10*(1), 439–446. https://journal.r-project.org/archive/2018/RJ-2018-009/index.html

Pebesma, E., & Bivand, R. S. (2005). Classes and methods for spatial data in R. *R News*, *5*(2), 9–13. https://journal.r-project.org/articles/RN-2005-014

QGIS Contributors. (2022). QGIS. *Zenodo*. https://doi.org/10.5281/zenodo.5869838

Seidel, D. P., Dougherty, E., Carlson, C., & Getz, W. M. (2018). Ecological metrics and methods for GPS movement data. *International Journal of Geographical Information Science : IJGIS*, *32*(11), 2272. https://doi.org/10.1080/13658816.2018.1498097

Shi, H., Huang, H., Ma, D., Chen, L., & Zhao, M. (2023). Capturing urban recreational hotspots from GPS data: A new framework in the lens of spatial heterogeneity. *Comput. Environ. Urban Syst.*, *103*, 101972. https://doi.org/10.1016/j.compenvurbsys.2023.101972

Thierry, B., Chaix, B., & Kestens, Y. (2013). Detecting activity locations from raw GPS data: a novel kernel-based algorithm. *Int. J. Health Geographics*, *12*, 14. https://doi.org/10.1186/1476-072X-12-14

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Mulbregt, P. van. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

WebAssembly Core Specification. (2019). https://www.w3.org/TR/wasm-core-1

Xu, Q., Xiang, L., Wu, H., Tao, C., Wang, H., Yu, L., Liu, Q., & Wang, X. (2024). Dynamic heatmap pyramid computation for massive high-parallel spatial streaming in urban environments. *Int. J. Digital Earth*. https://www.tandfonline.com/doi/full/10.1080/17538947.2024.2368099

Zhang, S., Nolting, D., & Das, A. (2023). Heat marks the spot: De-anonymizing user's geographical data on the strava heatmap. *7th Workshop on Technology and Consumer Protection (ConPro)*.