

# GTFS Segments: A fast and efficient library to generate bus stop spacings

Saipraneeth Devunuri<sup>1\*</sup> and Lewis Lehe<sup>1\*</sup>

<sup>1</sup> Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign ¶  
Corresponding author \* These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: ↗

Submitted: 28 September 2023

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

The GTFS Segments (`gtfs-segments`) library is an open-source Python toolkit for computing and analyzing bus stop spacings: the distance a transit bus travels between stops. The library reads General Transit Feed Specification (GTFS) data, snaps bus stops to points along routes, divides routes into segments (the pieces of routes between two stops), and then produces a `GeoDataFrame` containing information about each segment. The library also features a number of functions that act on this `GeoDataFrame`. It can produce summary statistics of the spacings for a given network, using various weighting schemes (i.e., weighting by frequency of service), as well as histograms of spacings that display their full distribution. In addition to network-level statistics, the package can also compute statistics for each route, such as its length, headway, speed, and average stop spacing. It can draw maps of networks, routes, or segments over a basemap—which allows the user to manually validate data. The segments data can be exported to `.csv` and `.geojson` files. The package can fetch the most up-to-date GTFS data from Mobility Data ([MobilityData, 2023](#)) repositories for user convenience.

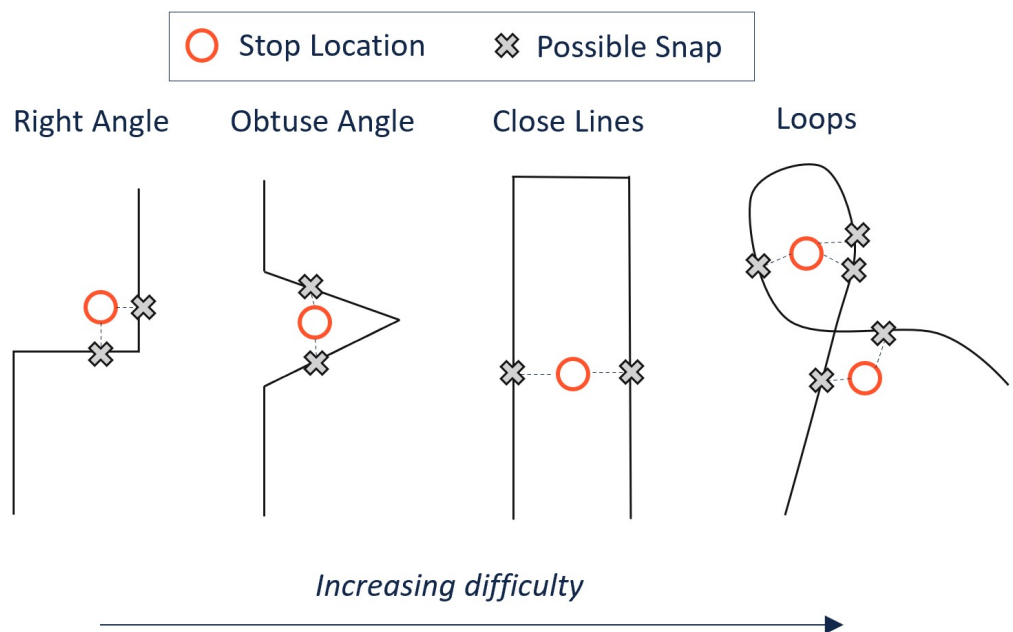
## Statement of need

The choice of bus stop spacing involves a tradeoff between accessibility and speed: wider spacings mean passengers must travel farther to/from stops, but they allow the bus to move faster ([Wu et al., 2022](#)). Many US transit agencies have recently carried out *stop consolidation* campaigns which systematically remove stops, due partly to the perception US stop spacings are much narrower than those abroad. However, there are no reliable data sources to obtain current stop spacings despite wide adoption of General Transit Feed Specification (GTFS) ([Voulgaris & Begwani, 2023](#)), because GTFS does not include data on stop spacings directly. Spacings must be computed from route shape geometries, stop locations, and stop sequences. A challenge is that stop locations are not placed on top of route shapes and therefore must be somehow projected onto the route's `LINestring`. To make spacings available for analysis, `gtfs-segments` uses *k*-dimensional spatial trees and *k*-nearest neighbor heuristics to snap stops to routes and divide routes into segments for computation of spacings, as described below.

`gtfs-segments` was designed for researchers, transit planners, students and anyone interested in bus networks. The package has been used in several scholarly articles ([Devunuri, Qiam, Lehe, Pandey, et al., 2023](#); [Devunuri, Qiam, & Lehe, 2023](#); [Lehe & Pandey, n.d.](#)) and to create databases of spacings for over 550 agencies in the US ([Devunuri et al., 2022](#)) and 80 agencies in Canada ([Devunuri, 2023](#)). Several transit agencies, such as Regional Transportation District Denver (RTD- Denver), have used the package to visualize the effects of their bus stop consolidation efforts. Filtering functions allow the user to explore datasets, identify errors and compute specialized statistics.

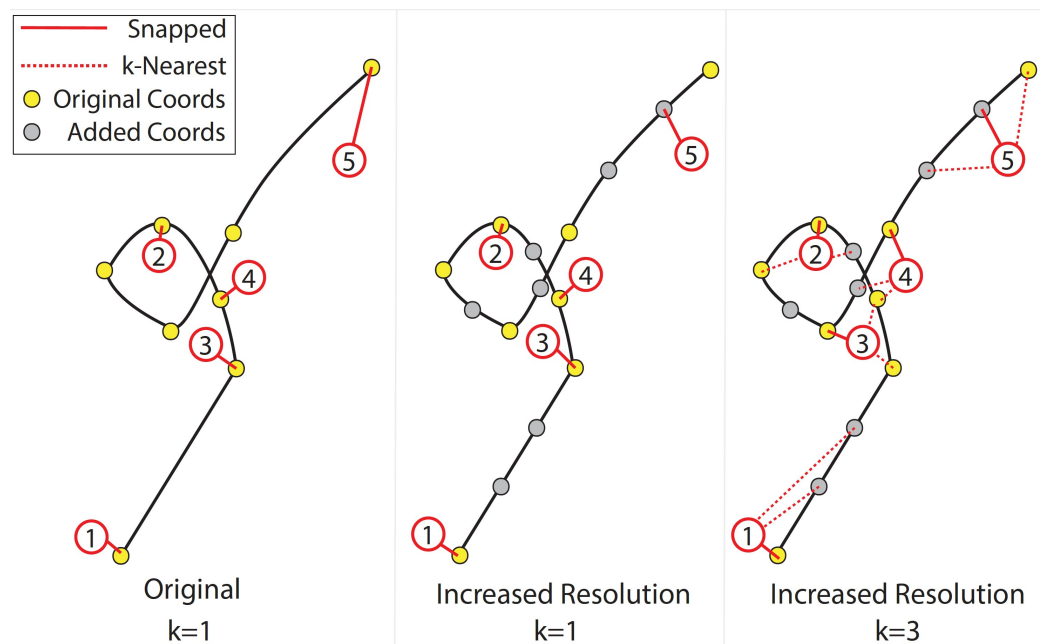
## 41 Computing Segments

42 A segment is defined by three elements: (i) a start stop, (ii) end stop and (iii) the path that the  
43 bus travels along the route in between the two consecutive stops. The segments are computed  
44 using route shape geometries and stop locations included in GTFS data. Packages such as  
45 `gtfs2gps` (Pereira et al., 2023) and `gtfs_functions` (Toso, 2023) can compute segments, but  
46 this package does so in computationally-efficient way and includes a new way to account  
47 for cases when the reported stop locations are misaligned with the route shapes. Figure 1  
48 provides some example cases where a stop is equidistant from multiple points on a route. Thus,  
49 projecting the stop onto the route or snapping to nearest geo-coordinate (lat,lon) may produce  
50 errors, such as stops that are out-of-order or stop being snapped far from their locations. Also,  
51 the time complexity of projection or snapping is  $O(mn)$  using brute force for  $m$  geo-coordinates  
52 that represent the route shape and  $n$  stops that have to be projected.



**Figure 1:** Example route shapes with stop locations that are equidistant from multiple points along the route.

53 `gtfs-segments` overcomes these challenges by increasing the route resolution (i.e., adding  
54 points in-between geo-coordinates), using spatial k-d trees, and using more than one nearest  
55 neighbor. The increase in resolution allows stops to be snapped to nearby points. k-d  
56 trees (Maneewongvatana & Mount, 1999) reduce the time complexity to  $O(n \log(m))$  and  
57 make it possible to compare among several snapping points without added computation.  
58 Figure 2 shows a difficult example route. In the first panel, snapping to the nearest point  
59 produces out-of-order stops (3/4/2) and stop 5 is snapped far away from its location. In the  
60 second panel, increased resolution fixes 5's location problem but the ordering problem persists.  
61 In the third panel, we use the  $k=3$  nearest neighbors and thus find a proper ordering. Once  
62 every stop has been snapped to a geo-coordinate on the route shape, the shape is segmented  
63 between stops and each segment represented by a `LINESTRING` for entry in the `GeoDataFrame`.  
64 In fact, `gtfs_segments` starts with  $k=3$  for the neighbors and doubles  $k$  until we find the correct  
65 sequence of stops or remove the corresponding trip. On average, fewer than 1% of trips fail,  
66 which can be manually corrected and validated.



**Figure 2:** Improvement in snapping due to an increase in resolution and using k-nearest neighbors.. Adapted from “Bus Stop Spacings Statistics: Theory and Evidence” (Devunuri, Qiam, Lehe, Pandey, et al., 2023)

## Acknowledgements

The gtfs-segments package draws its inspiration from gtfs\_functions(Toso, 2023), gtfs2gps(Pereira et al., 2023), and partridge(Whalen, 2023) repositories. We thank the contributors of these packages for their excellent work. We also extend our thanks to Mobility Data(MobilityData, 2023) for compiling GTFS from around the globe and constantly maintaining them.

## References

- Devunuri, S. (2023). *Bus Stop Spacings for Transit Providers in Canada* (Version V2) [Data set]. Harvard Dataverse. <https://doi.org/10.7910/DVN/QFTAPM>
- Devunuri, S., Qiam, S., & Lehe, L. (2022). *Bus Stop Spacings for Transit Providers in the US* (Version V3) [Data set]. Harvard Dataverse. <https://doi.org/10.7910/DVN/SFBIVU>
- Devunuri, S., Qiam, S., & Lehe, L. (2023). *ChatGPT for GTFS: From words to information*. <https://doi.org/10.48550/arXiv.2308.02618>
- Devunuri, S., Qiam, S., Lehe, L., Pandey, A., & Monzer, D. (2023). *Bus stop spacings statistics: Theory and evidence*. <https://doi.org/10.48550/arXiv.2208.04394>
- Lehe, L., & Pandey, A. (n.d.). Bus stop spacing with heterogeneous trip lengths and elastic demand. Available at SSRN 4135394. <https://doi.org/10.2139/ssrn.4135394>
- Maneewongvatana, S., & Mount, D. M. (1999). Analysis of approximate nearest neighbor searching with clustered point sets. *arXiv Preprint Cs/9901013*. <https://doi.org/10.48550/arXiv.cs/9901013>
- MobilityData. (2023). *Mobility Database*. <https://database.mobilitydata.org/>

- 88 Pereira, R. H., Andrade, P. R., & Vieira, J. P. B. (2023). Exploring the time geography of  
89 public transport networks with the gtfs2gps package. *Journal of Geographical Systems*,  
90 25(3), 453–466. <https://doi.org/10.31235/osf.io/qydr6>
- 91 Toso, S. (2023). *Gtfs\_functions: Package with useful functions to create geo-spatial visualiza-*  
92 *tions from a GTFS*. GitHub. [https://github.com/Bondify/gtfs\\_functions](https://github.com/Bondify/gtfs_functions)
- 93 Voulgaris, C. T., & Begwani, C. (2023). Predictors of early adoption of the general transit  
94 feed specification. *Findings*. <https://doi.org/10.32866/001c.57722>
- 95 Whalen, D. (2023). *Partridge: A fast, forgiving GTFS reader built on pandas DataFrames*.  
96 GitHub. <https://github.com/remix/partridge>
- 97 Wu, T., Jin, H., & Yang, X. (2022). To what extent may transit stop spacing be increased  
98 before driving away riders? Referring to evidence of the 2017 NHTS in the united states.  
99 *Sustainability*, 14(10). <https://doi.org/10.3390/su14106148>

DRAFT