The Journal of Open Source Software

PointClouds.jl: Fast & flexible processing of lidar data

Manuel F. Schmid [1], **Jeffrey D. Massey** [2], **and Marco G. Giometto** [1,2]

1 Columbia University, New York, NY, USA **2** Amazon, Seattle, WA, USA

DOI: 10.21105/joss.07952

Software

- Review
- Repository
- Archive

Editor: Kristen Thyng

Reviewers:

- @evetion
- @atzberg

Submitted: 15 October 2024
Published: 08 July 2025

Summary

PointClouds.jl is a toolbox for working with point-cloud data that lets users implement complex point-processing workflows. Geospatial point-cloud data is usually obtained with a "lidar" instrument that scans the surface with high-frequency laser pulses, and large datasets covering entire countries are being made publicly available online. PointClouds.jl includes functionality to query such datasets, load and save data in the commonly used LAS/LAZ file format, and extract useful information through a series of processing steps that extend, filter, or transform the point-cloud data.

Statement of need

Point-cloud data has found its way into numerous applications in industry, government, and academia. Examples include computer vision for robotics and autonomous vehicles, generation of digital terrain & surface models, forestry and agriculture, archaeological surveys, geomorphology, and inspection/surveillance of infrastructure. Thanks to falling acquisition costs and newly emerging sensor platforms, point-cloud data is more readily available than ever before.

Most applications rely on a series of automated processing steps to derive actionable information from the raw point-cloud data. These steps may include noise removal, subsetting, classification, object detection, rasterization, triangulation, and conversion between various data formats. Some processing steps may also make use of additional data obtained from complementary sensing technologies such as multispectral cameras or rely on pre-existing databases of geospatial information.

Various tools exist to aid with this data processing, both as standalone software and as add-ons to geospatial software. With PointClouds.jl we present a new contribution to this ecosystem that was motivated by four key requirements:

- interactive development of custom processing steps with solid performance,
- programmatic access to national databases of airborne lidar data,
- easy integration with other Julia software, and
- open-source availability of the code.

We expect that many applications within and outside of academia may have similar needs and therefore benefit from the approach we took with PointClouds.jl. In particular, we hope to increase adoption of national lidar datasets, encourage people to run their models on many locations rather than a few hand-picked ones, and promote more activity in the development of point-cloud processing techniques. PointClouds.jl is made available under the MIT license and relies on the Julia programming language (Bezanson et al., 2017) to run both included library functionality and custom code from the user with solid performance.

PointClouds.jl took shape as part of a research collaboration between Columbia University and Amazon that produced the above requirements, which were not fully met by existing solutions. The project is aimed at simulating near-surface wind dynamics in any populated area in the United States, with extensibility to other countries of interest. Such simulations could support drone-delivery operations in the future. PointClouds.jl allows us to construct a processing pipeline that can access the lidar data for the specified coordinates, derive rasterized representations of resolvable features (terrain, buildings), and estimate aerodynamic properties of unresolvable features (vegetation, surface roughness). Furthermore, it allows us to assess the sensitivity of wind predictions on properties of the input data and the point-cloud processing methods. Part of this work was presented at recent conferences (Giometto et al., 2024; Schmid et al., 2023; Schmid & Giometto, 2024) and further publications are in preparation.

Overview of functionality

The functionality of PointClouds.jl covers three main areas: data access, file input and output, and in-memory processing.

Data access: PointClouds.jl can query national lidar datasets using coordinates, automatically downloading the available data for the requested areas. Initial support is included for the USGS 3D Elevation Program (3DEP) in the United States whereas support for other national programs will be added over time.

File input and output: PointClouds.jl reads and writes the LAS format defined by the American Society for Photogrammetry and Remote Sensing (2019) and the compressed LAZ variant (Isenburg, 2013) in all current versions of the format (1.0 – 1.4) with strict adherence to the specification. It includes support for parsing the coordinate reference system (CRS) information and for working with files that do not fit into memory using lazy processing. While LAS/LAZ is the most common format for published point-cloud datasets, support for additional point-cloud formats is within the scope of PointClouds.jl and may be added over time, whereas support for general-purpose data formats such as HDF5 is delegated to separate Julia packages.

In-memory processing: PointClouds.jl provides robust fundamentals for multithreaded iteration over points, filtering, coordinate transforms, rasterization, and finding the nearest neighbors. Over time, PointClouds.jl is expected to grow a comprehensive library of point-processing algorithms that can be used as building blocks for complex spatial analysis tasks.

While we are not aware of any other software that is targeting programmatic access to national lidar datasets, there are quite a few existing tools that provide functionality for working with LAS data and applying various processing methods. The PDAL (Butler et al., 2021; PDAL Contributors, 2024) and lidR (Roussel et al., 2020) projects in particular provide functionality for geospatial point-cloud processing that is similar to the scope of PointClouds.jl. These packages, written in C++ and R respectively, are more mature and implement various processing steps ("filters" in PDAL, "metrics" in lidR) that are not yet available in PointClouds.jl. Applications that heavily rely on existing building blocks to set up a point-processing pipeline may therefore be better served by those alternatives, while we believe that PointClouds.jl is already a competitive offer for implementing custom processing steps, or when working with point-cloud data within a Julia project. Over time, we hope that PointClouds.jl can also match the maturity and comprehensiveness of the alternative choices.

There are also a number of existing Julia packages that share parts of the functionality and goals of PointClouds.jl. LAS/LAZ file support was pioneered by LasIO.jl and LazIO.jl, inspiring the corresponding functionality in PointClouds.jl, whereas LASDatasets.jl represents a recent effort for more comprehensive LAS/LAZ functionality. In the domain of geospatial data processing, the JuliaGeo and JuliaEarth ecosystems provide a variety of tools, many of which may be applied to point-cloud data. A few packages such as PointCloudRasterizers.jl are geared specifically towards such applications. Nevertheless, we see benefit from a project that optimizes the functionality, ergonomics, and performance across the entire process of finding, loading, and transforming point-cloud data.

Acknowledgements

We acknowledge the financial support from Amazon. Marco G. Giometto holds concurrent appointments as Assistant Professor at Columbia University and as Amazon Visiting Academic. This paper describes work performed at Columbia University and is not associated with Amazon. This material is based upon work supported by, or in part by, the Army Research Laboratory and the Army Research Office under grant number W911NF-22-1-0178.

References

American Society for Photogrammetry and Remote Sensing. (2019). *LAS specification 1.4 – R15*. https://publicdocuments.asprs.org/las-v14-r15-2019

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Butler, H., Chambers, B., Hartzell, P., & Glennie, C. (2021). PDAL: An open source library for the processing and analysis of point clouds. *Computers & Geosciences*, *148*, 104680. https://doi.org/10.1016/j.cageo.2020.104680

Giometto, M. G., Schmid, M. F., & Massey, J. D. (2024). *Automated processing of LiDAR data for high-fidelity flow modeling*. Presented at the AMS Annual Meeting, 28 Jan–1 Feb, Baltimore, MD; American Meteorological Society.

Isenburg, M. (2013). LASzip. *Photogrammetric Engineering & Remote Sensing*, *79*(2), 209–217. https://doi.org/10.14358/PERS.79.2.209

PDAL Contributors. (2024). *PDAL point data abstraction library*. Zenodo. https://doi.org/10.5281/zenodo.2616780

Roussel, J.-R., Auty, D., Coops, N. C., Tompalski, P., Goodbody, T. R. H., Meador, A. S., Bourdon, J.-F., de Boissieu, F., & Achim, A. (2020). lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. *Remote Sensing of Environment*, *251*, 112061. https://doi.org/10.1016/j.rse.2020.112061

Schmid, M. F., & Giometto, M. G. (2024). *Towards an integrated model for operational microscale simulations*. Presented at the AGU Annual Meeting, 9–13 Dec, Washington, D.C. American Geophysical Union.

Schmid, M. F., Massey, J. D., & Giometto, M. G. (2023). *Automated processing of LiDAR data for high-fidelity flow modeling*. Presented at the AGU Annual Meeting, 11–15 Dec, San Francisco, CA; American Geophysical Union.