

PyMultiLidar: A python package for lidar datasets with multiple point clouds

Thomas Goelles^{*1}, Stefan Muckenhuber^{1,2}, Birgit Schlager¹, Sarah Haas¹, and Tobias Hammer¹

1 Virtual Vehicle Research GmbH, Inffeldgasse 21A, 8010 Graz, Austria **2** University of Graz, Heinrichstrasse 36, 8010 Graz, Austria

DOI: [DOIunavailable](#)

Software

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

Editor: [Pending Editor](#) ↗

Reviewers:

- [@Pending Reviewers](#)

Submitted: N/A

Published: N/A

License

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

Summary

A lidar (light detection and ranging) is an active perception sensor operating in the optical or infrared part of the electromagnetic spectrum. Lidar sensors use the time-of-flight principle to measure the distance between sensor and illuminated object, i.e. a light pulse is sent out, reflected at an object, and the travel time between transmission and reception of the light pulse is measured to calculate the distance. Sending several light pulses in different directions and/or applying an array of receivers pointing in different directions, allows to create a 3D (or depth) image of the surrounding environment. Such a 3D image is typically stored as point cloud, where each point has assigned a distance and angle relative to the sensor and additional optional parameters such as intensity or reflectance. Recording multiple point clouds, rather than a single acquisition, from different locations allows to create a more complete 3D representation of the surrounding environment, since objects can then be illuminated from different angles. Recording multiple point clouds from the same locations, allows to investigate dynamic changes of the environment. Importing, processing, analysis, comparison, and visualization of multiple point clouds requires a well-defined data structure, relative positioning, orientation and timing information, and specifically designed, efficient tools and data pipelines. PyMultiLidar provides all these functionalities for lidar datasets consisting of multiple point clouds.

Statement of need

Considering recently emerging, promising lidar technologies, such as micro-electro-mechanical systems, optical phased array, vertical-cavity surface-emitting laser, single photon avalanche diode etc., combined with large efforts invested in particular by the automotive industry Thakur (2016) to further develop low-cost lidar systems, lidar sensors have the potential to enable a new cost-efficient way to perceive and measure the environment. This will not only have a strong impact on automotive applications but bears also large potential for other research fields and application domains, such as robotics, geophysics, etc. Already today, state-of-the-art lidar sensors designed for automotive applications, such as the Ouster OS-1 [<https://ouster.com>] or the Velodyne Ultra Puck [<https://velodynelidar.com>], offer many advantages: they are small in size, light in weight, robust, have a low eye safety class, and support high scanning speed. The expected substantial decrease in costs and increase in performance in the upcoming years will open up many new application areas for lidar systems.

PyMultiLidar aims to support testing and development of new application areas for lidar sensors by providing an open-source, user-friendly data processing tool for multiple lidar point clouds. PyMultiLidar is a python package for importing, processing, visualizing, and exporting lidar point clouds with focus on applications for multiple point clouds.

^{*}corresponding author Thomas.Goelles@v2c2.at

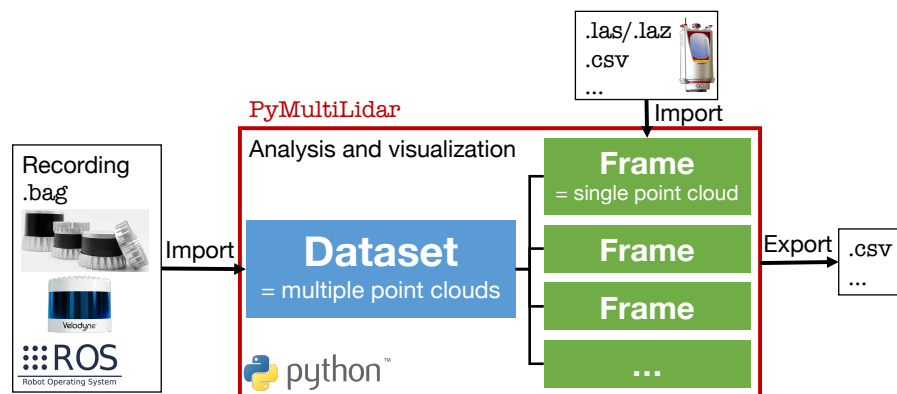


Figure 1: Data pipeline for lidar point clouds. The red box illustrates the functionality of PyMultiLidar. figure.

Figure 1 illustrates the overall lidar data pipeline of PyMultiLidar including import, export possibilities and the two main PyMultiLidar classes Dataset and Frame, which handle multiple and single point clouds respectively. Low-cost lidar systems typically support point cloud recording using ROS (robot operating system) (Stanford Artificial Intelligence Laboratory et al., 2018). PyMultiLidar can directly import multiple point clouds from ROS. Single point clouds can also be imported in other common formats, such as .las or .csv. PyMultiLidar builds on several well established python libraries and packages for data processing and visualization, such as numpy (Harris et al., 2020), pandas McKinney (2010), rospy (Stanford Artificial Intelligence Laboratory et al., 2018), open3D (Zhou et al., 2018), and plotly (Inc., 2015).

Acknowledgements

The publication was written at Virtual Vehicle Research GmbH in Graz, Austria. The authors would like to acknowledge the financial support within the COMET K2 Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action (BMK), the Austrian Federal Ministry for Digital and Economic Affairs (BMDW), the Province of Styria (Dept. 12) and the Styrian Business Promotion Agency (SFG). The Austrian Research Promotion Agency (FFG) has been authorised for the programme management.

References

- Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., R'io, J. F. del, Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hecht, J. (2018). Lidar for Self-Driving Cars. *Optics & Photonics News*, 29(1), 26. <https://doi.org/10.1364/OPN.29.1.000026>
- Inc., P. T. (2015). *Collaborative data science*. Plotly Technologies Inc. <https://plot.ly>
- McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). <https://doi.org/10.25080/Majora-92bf1922-00a>

- Stanford Artificial Intelligence Laboratory et al. (2018). *Robotic operating system (ROS Melodic Morenia)* [Computer software]. <https://www.ros.org>
- team, T. pandas development. (2020). *Pandas-dev/pandas: pandas (latest)* [Computer software]. Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- Thakur, R. (2016). Scanning LIDAR in advanced driver assistance systems and beyond: Building a road map for next-generation LIDAR technology. *IEEE Consumer Electronics Magazine*, 5(3), 48–54. <https://doi.org/10.1109/MCE.2016.2556878>
- Warren, M. E. (2019). Automotive LIDAR technology. *2019 Symposium on VLSI Circuits*, C254–C255. <https://doi.org/10.23919/VLSIC.2019.8777993>
- Zhou, Q.-Y., Park, J., & Koltun, V. (2018). Open3D: A modern library for 3D data processing. *arXiv:1801.09847*.