The Journal of Open Source Software

pytransform3d: 3D Transformations for Python

Alexander Fabisch[1]

1 Robotics Innovation Center, DFKI GmbH

DOI: 10.21105/joss.01159

Software

- Review
- Repository
- Archive

Submitted: 07 December 2018
Published: 31 January 2019

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

Summary

pytransform3d is a Python library for transformations in three dimensions. Heterogenous software systems that consist of proprietary and open source software are often combined when we work with transformations. For example, suppose you want to transfer a trajectory demonstrated by a human to a robot. The human trajectory could be measured from an RGB-D camera, fused with inertial measurement units that are attached to the human, and then translated to joint angles by inverse kinematics. That involves at least three different software systems that might all use different conventions for transformations. Sometimes even one software uses more than one convention. The following aspects are of crucial importance to glue and debug transformations in systems with heterogenous and often incompatible software:

- Compatibility: Compatibility between heterogenous softwares is a difficult topic. It might involve, for example, communicating between proprietary and open source software or different languages.
- Conventions: Lots of different conventions are used for transformations in three dimensions. These have to be determined or specified.
- Conversions: We need conversions between these conventions to communicate transformations between different systems.
- Visualization: Finally, transformations should be visually verified and that should be as easy as possible.

pytransform3d assists in solving these issues. Its documentation clearly states all of the used conventions, it makes conversions between rotation and transformation conventions as easy as possible, it is tightly coupled with Matplotlib to quickly visualize (or animate) transformations and it is written in Python with few dependencies. Python is a widely adopted language. It is used in many domains and supports a wide spectrum of communication to other software.

The library focuses on readability and debugging, not on computational efficiency. If you want to have an efficient implementation of some function from the library you can easily extract the relevant code and implement it more efficiently in a language of your choice.

The library integrates well with the scientific Python ecosystem (G. Varoquaux, Gouillart, & Vahtras, 2018) with its core libraries Numpy, Scipy and Matplotlib. We rely on Numpy (Walt, Colbert, & Varoquaux, 2011) for linear algebra and on Matplotlib (Hunter, 2007) to offer plotting functionalities. Scipy (Jones, Oliphant, Peterson, & others, 2001–2001--) is used if you want to automatically compute new transformations from a graph of existing transformations.

More advanced features of the library are the TransformManager which manages complex chains of transformations, the TransformEditor which allows to modify transformations

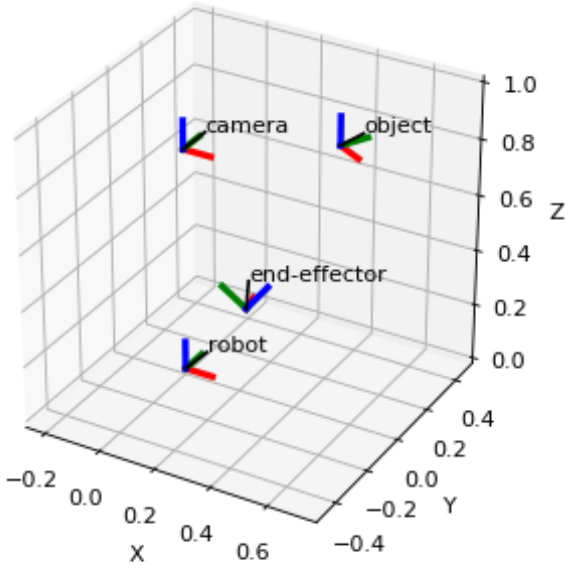

Figure 1: Transformation Manager: The TransformManager builds a graph of transformations that can be used to automatically infer previously unknown transformations.

graphically (additionally requires PyQt4), and the UrdfTransformManager which is able to load transformations from the Unified Robot Description Format (URDF) (additionally requires beautifulsoup4).

One of the strengths of pytransform3d in comparison to most other libraries is its rigorous approach to testing. Unit tests have 100% branch coverage for code that is not related to visualization, there is more test code than library code, there are additional examples, and continuous integration runs with Python 2.7, 3.4, 3.5, and 3.6. The maintainer ensures that this level of quality will not be sacrificed for new features.

There are several similar software packages. ROS tf2 (Foote, Marder-Eppstein, Meeussen, & others, 2009–2009--) is probably the most widely used of them. It offers functionality that is similar to the TransformManager of pytransform3d, but also considers temporal aspects of transformations. It also provides conversions between various conventions and visualization can be done with ROS' rviz package. EnviRe (Carrió et al., 2016) provides similar functionality. However, both libraries come with a number of dependencies and require complex build tools. Hence, it is not easy to set them up quickly in a new environment with minimum impact to the system, while pytransform3d is a lightweight library that only requires the basic scientific Python software stack that runs on any machine that supports CPython. There are other lightweight Python libraries that offer transformations and conversions between conventions, for example, transforms3d (Brett & Gohlke, 2009–2009--) and rowan (Ramasubramani & Glotzer, 2018), but these do not directly support visualization.

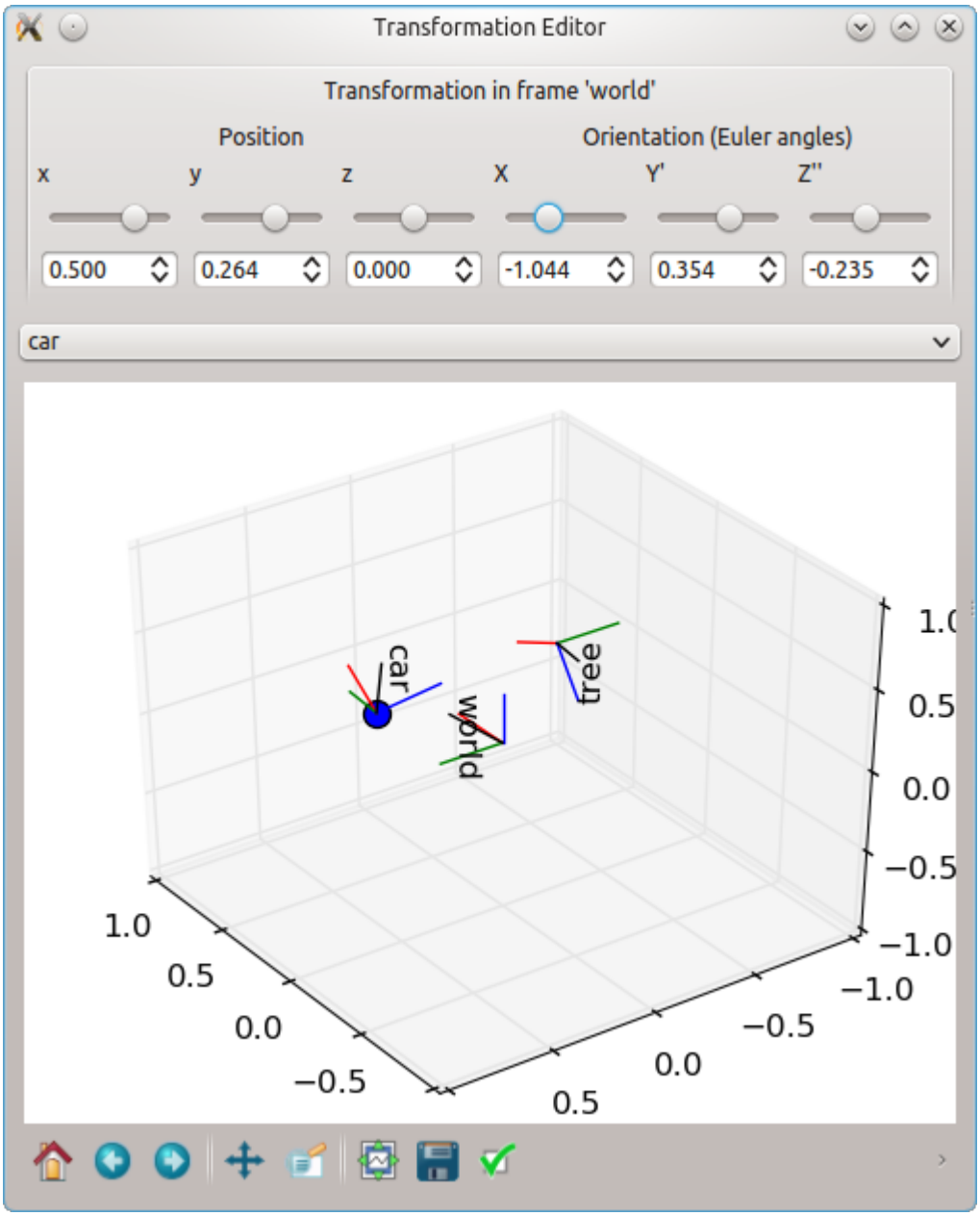

Figure 2: Transformation Editor: The TransformEditor based on PyQt4 can be used to visually modify transformations with a minimal number dependencies.

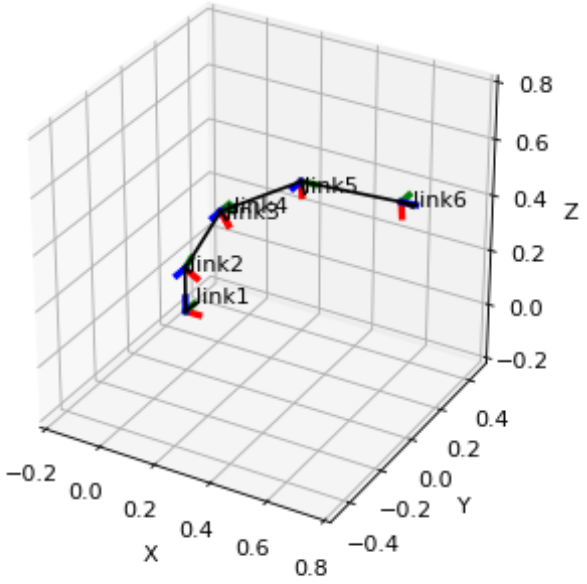

Figure 3: A simple URDF file loaded with pytransform3d and displayed in Matplotlib.

Research

pytransform3d is used in various domains, for example, specifying motions of a robot, learning robot movements from human demonstration, and sensor fusion for human pose estimation. pytransform3d has been used by Gutzeit et al. (2018) in the context of learning robot behaviors from demonstration.

Acknowledgements

We would like to thank Manuel Meder and Hendrik Wiese who have given valuable feedback as users to improve pytransform3d. This work was supported through two grants of the German Federal Ministry of Economics and Technology (BMWi, FKZ 50 RA 1216 and FKZ 50 RA 1217) and two grants from the European Union's Horizon 2020 research and innovation program (723853 and 730014).

References

Brett, M., & Gohlke, C. (2009–2009--). Transforms3d. Retrieved from https://github.com/matthew-brett/transforms3d

Carrió, J. H., Arnold, S., Böckmann, A., Born, A., Domínguez, R., Hennes, D., Hertzberg, C., et al. (2016). EnviRe – environment representation for long-term autonomy. In *AI for long-term autonomy workshop on the international conference on robotics and automation (icra)*.

Foote, T., Marder-Eppstein, E., Meeussen, W., & others. (2009–2009--). ROS tf2. Retrieved from http://wiki.ros.org/tf2

Gutzeit, L., Fabisch, A., Otto, M., Metzen, J. H., Hansen, J., Kirchner, F., & Kirchner, E. A. (2018). The besman learning platform for automated robot skill learning. *Frontiers in Robotics and AI*, *5*, 43. doi:10.3389/frobt.2018.00043

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Jones, E., Oliphant, T., Peterson, P., & others. (2001–2001--). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/

Ramasubramani, V., & Glotzer, S. C. (2018). Rowan: A python package for working with quaternions. *Journal of Open Source Software*, *3*(27). doi:10.21105/joss.00787

Varoquaux, G., Gouillart, E., & Vahtras, O. (2018). Scipy lecture notes. Retrieved from https://www.scipy-lectures.org/

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37