

OmniCV - A library for Omnidirectional Cameras

Kaustubh Sadekar¹, Leena Vachhani¹, and Abhishek Gupta²

¹ IDP in Systems and Control Engineering, IIT Bombay, India ² Department of Mechanical Engineering, IIT Bombay, India

DOI: [10.21105/joss.02533](https://doi.org/10.21105/joss.02533)

Software

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

Editor: [Matthew Sottile](#) ↗

Reviewers:

- [@nnadeau](#)
- [@ionlights](#)

Submitted: 02 July 2020

Published: 28 July 2020

License

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

Summary

Omnidirectional cameras refer to cameras with large field-of-view (FOV), more than 180°, and ideally 360°. This includes cameras with horizontal 360° field-of-view(FOV) and cameras with FOV spanning a 360° horizontal and more than 90 (ideally 180) vertical field-of-view(FOV). It is beneficial to use such large field-of-view cameras, especially for indoor environments (Scaramuzza (2014)). There has been a constant growth in research related to omnidirectional cameras (Zhang, Rebecq, Forster, & Scaramuzza (2016)). Most of the applications using such cameras either need a mathematical model of the camera or some form of mapping to represent the 360° image information like equirectangular projection or cubemap projection.

OmniCV is a complete library that is written to support and enhance the research related to omnidirectional cameras. It contains, different class methods and APIs to, simulate virtual cameras based on different omnidirectional camera models and also contains methods for interconversion of different representations of a 360° image like equirectangular projection, cubemap projection and perspective projection. The library also contains some essential GUI based software written using the package APIs that enable the user to determine various parameters of the camera used to capture any given image. All the functionalities are written in C++ as well as python. The operations are optimized to ensure significant use in real-time applications. Moreover, ROS nodes for all the functionalities are also provided for easy integration with any existing ROS project.

Several applications and research projects related to omnidirectional cameras have used a few of the above-mentioned functionalities and have also open-sourced their code but there is no single complete open-source library available with C++, python and ROS compatibility. In “DeepCalib: A deep learning approach for automatic intrinsic calibration of wide field-of-view cameras” (Bogdan, Eckstein, Rameau, & Bazin (2018)), a virtual camera based on the unified spherical model (Barreto (2006)) is used to generate multiple synthetic images, with different intrinsic parameters, for each 360° image of the sun360 dataset. In “Flat2Sphere: Learning Spherical Convolution for Fast Features from 360° Imagery” (Su & Grauman (2017)) inverse perspective projection to equirectangular projections at different polar angles is used to test the model on PASCAL VOC dataset. In “CubemapSLAM: A Piecewise-Pinhole Monocular Fisheye SLAM System” (Wang et al. (2018)) the SLAM problem for a fisheye camera is addressed by converting the fisheye camera image to cubemap representation. There are some open-source projects like (“Fisheyedwarp,” n.d.) and (“Py360convert,” n.d.) which provide python-based implementations for a few interconversion of 360° images from one format to the other but there is no such complete library which provides different camera models, ROS nodes and optimized implementations for various forms of interconversions of 360° images with C++ as well as Python support. Several omnidirectional camera models like Unified Camera Model (Geyer & Daniilidis (2000)), Extended Unified Camera Model (Khomutenko, Garcia, & Martinet (2016)), Field-Of-View camera model(Devernay & Faugeras (2001)) as well as Double Sphere camera model (Usenko, Demmel, & Cremers (2018)) have been implemented in OmniCV to support research related to omnidirectional cameras using multiple types of camera models rather than depending on just a single model.

OmniCV library has been developed with the following objectives: 1) Quick and easy to use API to encourage and enhance the research in areas using omnidirectional cameras. 2) To support real-time applications. 3) Provide extensions in python as well as C++ as they are the languages used by researchers. 4) Provide a ROS package to use in robotics research. Detailed documentation is also provided, with stepwise instructions to install the library and detailed examples for each function are provided to make it easy to understand and use. A modular object-oriented programming method is used for the library, making it easily compatible and manageable for future updates.

Acknowledgments

This work has made use of the following softwares : numpy (Walt, Colbert, & Varoquaux (2011)) , matplotlib ((???)), jupyter (Kluyver et al. (2016)) and OpenCV ((“Open Source Computer Vision Library,” 2015)). Images from the fisheye dataset(Eichenseer & Kaup (2016)) were used in the experiments related to fisheye images.

This work is funded by a project under the National Center of Excellence in Technology and Internal Security (NCETIS), IIT Bombay sponsored by the Ministry of Electronics and Information Technology.

References

- Barreto, J. P. (2006). A unifying geometric representation for central projection systems. *Computer Vision and Image Understanding*, 103, 208–217. doi:[10.1016/j.cviu.2006.06.003](https://doi.org/10.1016/j.cviu.2006.06.003)
- Bogdan, O., Eckstein, V., Rameau, F., & Bazin, J.-C. (2018). DeepCalib: A deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In (pp. 1–10). doi:[10.1145/3278471.3278479](https://doi.org/10.1145/3278471.3278479)
- Devernay, F., & Faugeras, O. (2001). Straight lines have to be straight automatic calibration and removal of distortion from scenes of structured environments. *Mach Vis Appl*, 13. doi:[10.1007/PL00013269](https://doi.org/10.1007/PL00013269)
- Eichenseer, A., & Kaup, A. (2016). A data set providing synthetic and real-world fisheye video sequences. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1541–1545).
- Fisheyedwarp. (n.d.). <https://github.com/BlueHorn07/py-fisheye-dewarp>.
- Geyer, C., & Daniilidis, K. (2000). A unifying theory for central panoramic systems and practical implications. In *ECCV*. doi:[10.1007/3-540-45053-X_29](https://doi.org/10.1007/3-540-45053-X_29)
- Khomutenko, B., Garcia, G., & Martinet, P. (2016). An enhanced unified camera model. *IEEE Robotics and Automation Letters*, 1(1), 137–144.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks ? A publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press. Retrieved from <https://eprints.soton.ac.uk/403913/>
- Open Source Computer Vision Library. (2015). <https://opencv.org>.
- Py360convert. (n.d.). <https://github.com/sunset1995/py360convert>.
- Scaramuzza, D. (2014). Omnidirectional camera. In K. Ikeuchi (Ed.), *Computer vision: A reference guide* (pp. 552–560). Boston, MA: Springer US. doi:[10.1007/978-0-387-31439-6_488](https://doi.org/10.1007/978-0-387-31439-6_488)

- Su, Y.-C., & Grauman, K. (2017). Learning spherical convolution for fast features from 360Imagery. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 529–539). Curran Associates, Inc. Retrieved from <http://papers.nips.cc/paper/6656-learning-spherical-convolution-for-fast-features-from-360-imagery.pdf>
- Usenko, V., Demmel, N., & Cremers, D. (2018). The double sphere camera model. In *Proc. Of the int. Conference on 3D vision (3DV)*. Retrieved from <http://arxiv.org/abs/1807.08957>
- Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13, 22–30. doi:[10.1109/MCSE.2011.37](https://doi.org/10.1109/MCSE.2011.37)
- Wang, Y., Cai, S., Li, S.-J., Liu, Y., Guo, Y., Li, T., & Cheng, M.-M. (2018). CubemapSLAM: A piecewise-pinhole monocular fisheye slam system. Retrieved from <http://arxiv.org/abs/1811.12633>
- Zhang, Z., Rebecq, H., Forster, C., & Scaramuzza, D. (2016). Benefit of large field-of-view cameras for visual odometry. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 801–808.