The Journal of Open Source Software

IMPPY3D: Image Processing in Python for 3D Image Stacks

Newell H. Moser [1¶], **Alexander K. Landauer** [2], and **Orion L. Kafka** [1]

1 Material Measurement Laboratory, National Institute of Standards and Technology, 325 Broadway, Boulder, CO, 80305, USA **2** Material Measurement Laboratory, National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, 20899, MD, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @sitic
- @gknapp1

Submitted: 17 October 2024
Published: unpublished

Summary

Image Processing in Python for 3D image stacks, or IMPPY3D, is a free and open-source software (FOSS) repository that simplifies post-processing and 3D shape characterization for grayscale image stacks, otherwise known as volumetric images, 3D images, or voxel models. While IMPPY3D, pronounced impee-three-dee, was originally created for post-processing image stacks generated from X-ray computed tomography (XCT) measurements, it can be applied generally in post-processing 2D and 3D images. IMPPY3D includes tools for segmenting volumetric images and characterizing the 3D shape of features or regions of interest. These functionalities have proven useful in 3D shape analysis of powder particles, porous polymers, concrete aggregates, internal pores/defects, and more (see the Research Applications section). IMPPY3D consists of a combination of original Python scripts, Cython extensions, and convenience wrappers for popular third-party libraries like SciKit-Image, OpenCV, and PyVista (Bradski, 2000; Sullivan & Kaszynski, 2019; Walt et al., 2014).

Highlighted capabilities of IMPPY3D include: varying image processing parameters interactively, applying numerous 2D/3D image filters (e.g., blurring/sharpening, denoising, erosion/dilation), segmenting and labeling continuous 3D objects, precisely rotating and re-slicing an image stack in 3D, generating rotated bounding boxes fitted to voxelized features, converting image stacks into 3D voxel models, exporting 3D models as Visualization Toolkik (VTK) files for ParaView (Ayachit, 2015), and converting voxel models into smooth mesh-based models. Additional information and example scripts can be found in the included ReadMe files within the IMPPY3D GitHub repository (Moser, Landauer, et al., 2024). As a visualized example, Figure Figure 1 demonstrates the high-level steps to characterize powder particles using IMPPY3D. This workflow is also similar to how pores can be visualized and characterized in metal-based additive manufacturing. Additional research applications for IMPPY3D are discussed in a later section.

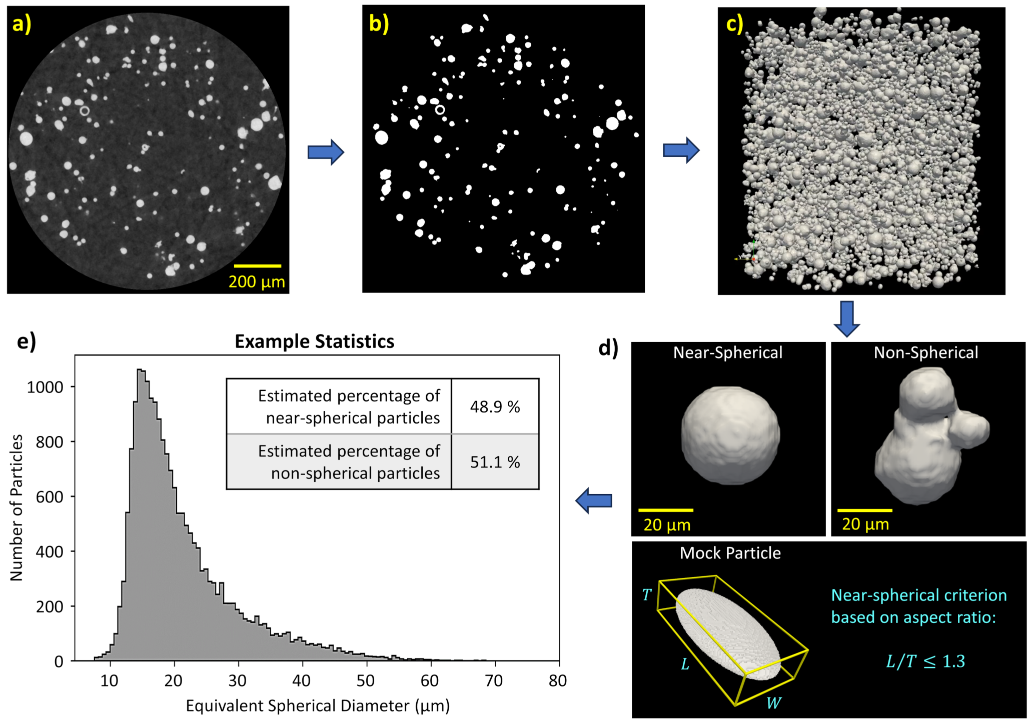

Figure 1: X-ray computed tomography reconstructions of nickel-based powder particles suspended in cured epoxy. a) One reconstructed 2D image slice (out of 1009) illustrating the powder particles, and b) the same image after segmentation using a series of filtering and binarization techniques. c) A rendering of the interactive 3D model of the segmented particle volume image. d) Individual particles visualized for characterization based on shape, volume, and porosity. e) The ratio of spherical to non-spherical particles and a histogram plot showing the distribution in size of the particles.

Statement of Need

Volumetric images commonly arise from nondestructive measurement techniques such as XCT, optical coherence tomography (OCT) or confocal microscopy, or from destructive techniques such as serial sectioning. Volumetric images typically analyzed in IMPPY3D are grayscale representations of a 3D volume of material and contain both internal and external shape information. For example, XCT is commonly used in metal-based additive manufacturing to prevent parts from entering service that contain critical internal defects. Post-reconstruction image analysis software is often employed to post-process volumetric images, such as Dragonfly[1] and Avizo (*Avizo*, 2024; *Dragonfly 3D World*, 2024). While closed-source software packages are highly sophisticated tools, they are also inherently limited since users are restricted by the closed-source publishing model. Users may require specific features that are unavailable, and closed-source models can be difficult or impossible to validate and verify.

Non-commercial software packages are also available that post-process volumetric images with varying degrees of generality and openness. While not an exhaustive list, examples include ImageJ/FIJI, 3D Slicer, DREAM.3D, SPAM, and PuMA (*3D Slicer*, 2024; Ferguson et al., 2021; Groeber & Jackson, 2014; Schindelin et al., 2012; Stamati et al., 2020). However, existing software can be difficult to extend for custom analyses, and/or their current features and strengths lie outside of volumetric segmentation and 3D shape characterization. There are

[1]Certain equipment, instruments, software, or materials are identified in this paper in order to specify the data processing procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

also FOSS packages that specialize in tomographic reconstruction, such as TomoPy and Tomviz (Gürsoy et al., 2014; *Tomviz*, 2024). However, the focus of IMPPY3D is the segmentation and feature analysis of already-reconstructed 3D image stacks, rather than image reconstruction itself. IMPPY3D is written in straightforward Python that contains internal documentation with the goal of making the library flexible and extensible to anyone with basic knowledge of Python and image processing. The library has been designed to work within an Conda/Miniforge environment for either Windows or Linux machines.

Research Applications of IMPPY3D

IMPPY3D has been in development since 2021. During this period, the library has evolved and been used in several research thrusts at the National Institute of Standards and Technology (NIST). Examples of published research applications, mostly related to XCT, include additive manufacturing, impact mitigating foams, powder particles, concrete aggregates, and lunar soil/regolith. A non-exhaustive list of publications involving IMPPY3D includes:

- Goguen et al. (2024), Three-dimensional characterization of particle size, shape, and internal porosity for Apollo 11 and Apollo 14 lunar regolith and JSC-1A lunar regolith soil simulant
- Moser, Benzing, et al. (2024), AM Bench 2022 Macroscale Tensile Challenge at Different Orientations (CHAL-AMB2022-04-MaTTO) and Summary of Predictions
- Kafka et al. (2024), A technique for in-situ displacement and strain measurement with laboratory-scale X-ray Computed Tomography
- Landauer, Kafka, et al. (2023), A materials data framework and dataset for elastomeric foam impact mitigating materials
- Landauer, Tsinas, et al. (2023), Unintended consequences: Assessing thermo-mechanical changes in vinyl nitrile foam due to micro-computed X-ray tomographic imaging
- Derimow et al. (2022), Surface globularization generated by standard PBF-EB Ti-6Al-4V processing achieves an improvement in fatigue performance

Getting Started with IMPPY3D

To begin using IMPPY3D, a Python environment with the necessary dependencies must be installed. We have deployed the code using the open-source package manager "Mamba" from Miniforge (version 24.3.0) based on Python 3.10 (*Miniforge*, 2024). The IMPPY3D GitHub repository (Moser, Landauer, et al., 2024) contains a dependencies folder which provides environment files (.yml) and a "ReadMe.txt" file that explains how to install a new Python environment using these environment files. In addition to "Mamba" (or "Conda" for Anaconda users), there are also generic instructions on how to install the necessary dependencies using PIP. Currently, IMPPY3D has been tested to work on modern Windows and Linux machines for Python versions 3.9 and 3.10. For users to test the success of the installation of the Python environment, there are example scripts in the "examples" folder in the IMPPY3D GitHub repository. These examples are also documented in a "ReadMe.txt" file.

In summary, IMPPY3D is a library of tools designed to accelerate the post-processing of image stacks. The package does not include a graphical user-interface (GUI). Therefore, users are expected to write their own Python scripts that utilize the IMPPY3D library, and the provided examples serve as templates that illustrate how to use a wide range of the functionality available in IMPPY3D. Typical processing pipeline options in IMPPY3D is illustrated in Figure Figure 2.

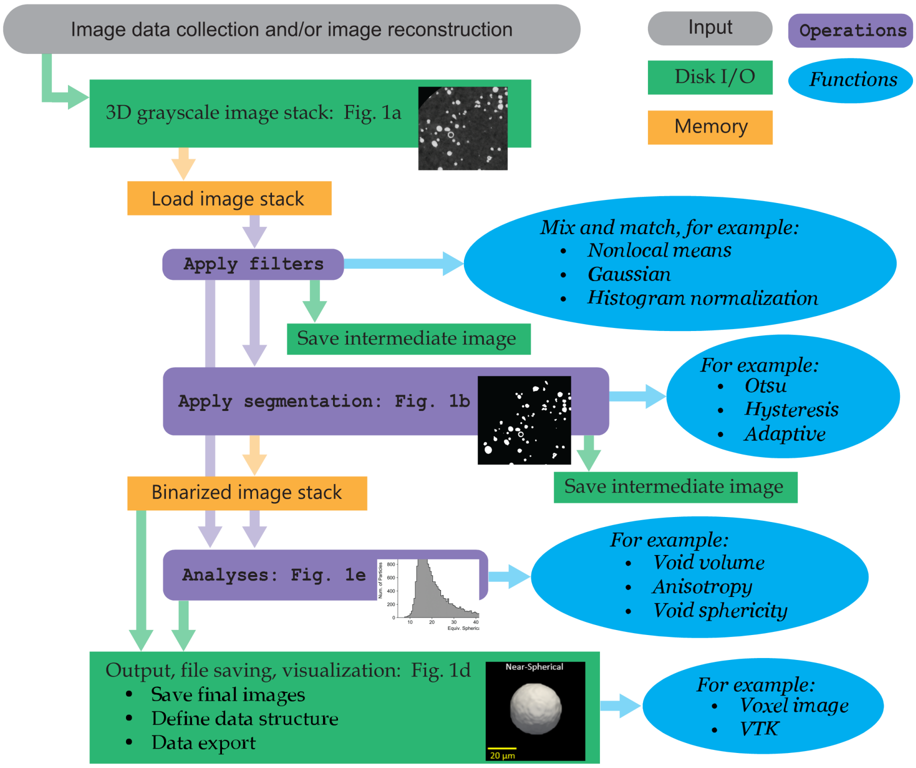

Figure 2: A high-level processing pipeline diagram illustrating typical steps and options available in IMPPY3D for 3D image stacks.

Acknowledgements

The authors would like to thank Dr. Edward J. Garboczi for his thought-provoking discussions and general guidance during the development of IMPPY3D. This research was partly performed while the authors each held a National Research Council Postdoctoral Research Associateship at the National Institute of Standards and Technology.

References

3D Slicer. (2024). Brigham and Women's Hospital. https://www.slicer.org/

Avizo. (2024). Thermo Fisher Scientific. https://www.thermofisher.com/us/en/home/electron-microscopy/products/software-em-3d-vis/avizo-software.html

Ayachit, U. (2015). *The ParaView guide: Updated for ParaView version 4.3* (L. Avila, Ed.; Full color version). Kitware Inc. ISBN: 978-1-930934-30-6

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Derimow, N., Hanson, K., Moser, N., Kafka, O. L., Benzing, J. T., & Hrabe, N. (2022). Surface globularization generated by standard PBF-EB Ti-6Al-4V processing achieves an improvement in fatigue performance. *International Journal of Fatigue*, *159*, 106810. https://doi.org/10.1016/j.ijfatigue.2022.106810

Dragonfly 3D World. (2024). Comet Group. https://dragonfly.comet.tech/en/product-overview

Ferguson, J. C., Semeraro, F., Thornton, J. M., Panerai, F., Borner, A., & Mansour,

N. N. (2021). Update 3.0 to "PuMA: The Porous Microstructure Analysis software," (PII:S2352711018300281). *SoftwareX*, *15*, 100775. https://doi.org/10.1016/j.softx.2021.100775

Goguen, J., Sharits, A., Chiaramonti, A., Lafarge, T., & Garboczi, E. (2024). Three-dimensional characterization of particle size, shape, and internal porosity for Apollo 11 and Apollo 14 lunar regolith and JSC-1A lunar regolith soil simulant. *Icarus*, *420*, 116166. https://doi.org/10.1016/j.icarus.2024.116166

Groeber, M. A., & Jackson, M. A. (2014). DREAM.3D: A Digital Representation Environment for the Analysis of Microstructure in 3D. *Integrating Materials and Manufacturing Innovation*, *3*(1), 56–72. https://doi.org/10.1186/2193-9772-3-5

Gürsoy, D., De Carlo, F., Xiao, X., & Jacobsen, C. (2014). TomoPy: A framework for the analysis of synchrotron tomographic data. *Journal of Synchrotron Radiation*, *21*(5), 1188–1193. https://doi.org/10.1107/S1600577514013939

Kafka, O. L., Landauer, A. K., Benzing, J. T., Moser, N. H., Mansfield, E., & Garboczi, E. J. (2024). A Technique for In-Situ Displacement and Strain Measurement with Laboratory-Scale X-Ray Computed Tomography. *Exp Tech*. https://doi.org/10.1007/s40799-024-00715-y

Landauer, A. K., Kafka, O. L., Moser, N. H., Foster, I., Blaiszik, B., & Forster, A. M. (2023). A materials data framework and dataset for elastomeric foam impact mitigating materials. *Scientific Data*, *10*(1), 356. https://doi.org/10.1038/s41597-023-02092-4

Landauer, A. K., Tsinas, Z., Kafka, O. L., Moser, N. H., Glover, J. L., & Forster, A. M. (2023). Unintended consequences: Assessing thermo-mechanical changes in vinyl nitrile foam due to micro-computed X-ray tomographic imaging. *Materials & Design*, *235*, 112381. https://doi.org/10.1016/j.matdes.2023.112381

Miniforge. (2024). https://github.com/conda-forge/miniforge

Moser, N., Benzing, J., Kafka, O. L., Weaver, J., Derimow, N., Rentz, R., & Hrabe, N. (2024). AM Bench 2022 Macroscale Tensile Challenge at Different Orientations (CHAL-AMB2022-04-MaTTO) and Summary of Predictions. *Integrating Materials and Manufacturing Innovation*. https://doi.org/10.1007/s40192-023-00333-3

Moser, N., Landauer, A., & Kafka, O. (2024). *IMPPY3D*. NIST. https://github.com/usnistgov/imppy3d

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., & Cardona, A. (2012). Fiji: An open-source platform for biological-image analysis. *Nature Methods*, *9*(7), 676–682. https://doi.org/10.1038/nmeth.2019

Stamati, O., Andò, E., Roubin, E., Cailletaud, R., Wiebicke, M., Pinzon, G., Couture, C., Hurley, R., Caulk, R., Caillerie, D., Matsushima, T., Bésuelle, P., Bertoni, F., Arnaud, T., Laborin, A., Rorato, R., Sun, Y., Tengattini, A., Okubadejo, O., … Birmpilis, G. (2020). Spam: Software for Practical Analysis of Materials. *Journal of Open Source Software*, *5*(51), 2286. https://doi.org/10.21105/joss.02286

Sullivan, C. B., & Kaszynski, A. (2019). PyVista: 3D plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *Journal of Open Source Software*, *4*(37), 1450. https://doi.org/10.21105/joss.01450

Tomviz. (2024). Kitware. https://tomviz.org/

Walt, S. van der, Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & contributors, the scikit-image. (2014). Scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453