

Tutorial on the Visualization of Volumetric Data Using *tomviz*

Barnaby D.A. Levin,^{1,2*} Yi Jiang,³ Elliot Padgett,¹ Shawn Waldon,⁴ Cory Quammen,⁴ Chris Harris,⁴ Utkarsh Ayachit,⁴ Marcus Hanwell,⁴ Peter Ercius,⁵ David A. Muller,^{1,6} and Robert Hovden⁷

¹School of Applied and Engineering Physics, 271 Clark Hall, Cornell University, Ithaca, NY 14853

²School for Engineering of Matter, Transport and Energy, 501 E. Tyler Mall, Arizona State University, Tempe, AZ 85287

³Department of Physics, 109 Clark Hall, Cornell University, Ithaca, NY 14853

⁴Kitware Inc., 28 Corporate Dr., #101, Clifton Park, NY 12065

⁵National Center for Electron Microscopy, Molecular Foundry, Lawrence Berkeley National Laboratory, One Cyclotron Rd., Berkeley, CA 94720

⁶Kavli Institute for Nanoscale Science, 420 Physical Sciences Building, Cornell University, Ithaca, NY 14853

⁷Department of Materials Science and Engineering, 2300 Hayward St., University of Michigan, Ann Arbor, MI 48109

*blevin2@asu.edu

Abstract: Tomography produces complex volumetric datasets containing the entire internal structure and density of an object in three dimensions (3D). Interpreting volumetric data requires 3D visualization but needs specialized software distinguishable from more familiar tools used in animation for 3D surface data. This tutorial reviews 3D visualization techniques for volumetric data using the open-source *tomviz* software package. A suite of tools including two-dimensional (2D) slices, surface contours, and full volume rendering provide quantitative and qualitative analysis of volumetric information. The principles outlined here are applicable to a wide range of 3D tomography techniques and can be applied to volumetric datasets beyond materials characterization.

Keywords: 3D visualization, tomography, electron microscopy, open science, *tomviz*

Introduction

This article is a short tutorial on the principles of visualizing complex 3D volumetric datasets, demonstrated using *tomviz*. Across a diverse range of scientific disciplines, understanding the 3D internal structure of material and biological specimens is essential for scientific progress. There are numerous methods for characterizing the 3D internal structure of objects at different length scales, including X-ray computed tomography [1], transmission electron microscopy (TEM) tomography [2,3], scanning transmission electron microscopy (STEM) tomography [4], focused ion beam-scanning electron microscopy (FIB-SEM) tomography [5], and atom-probe tomography [6,7]. Whatever the application, or the technique used, the volumetric datasets generated by these methods require visualization. Data visualization adds more than just aesthetic value to scientific research. High-quality and interpretable visualization is essential for extracting meaningful information from complex 3D structures.

Visualizing 3D volumetric datasets requires specialized software, distinguishable from the more familiar 3D visualization of wireframes (used in animated cinema) or molecular coordinates (crystallography). This specialized software requires interactive volume rendering tools, surface rendering tools, and the ability to display cross sections through the volume—at a minimum. Visualizations should be reproducible and shareable, enabling other scientists to inspect and validate data as well as understand the steps taken to turn the raw data

into the final result. Ultimately, the software should be able to produce publication quality figures for scientific articles.

The open-source *tomviz* software package aims to meet each of these criteria, whilst also being free to download and use. *Tomviz*, can be run on Windows, Mac OS, or Linux operating systems and is available for download at www.tomviz.org. *Tomviz* can be run on a laptop and can leverage powerful GPUs for visualizing large, intricate datasets. We use *tomviz* to illustrate this tutorial, but the principles outlined apply to other software packages, including the free non-commercial *Chimera* (University of California at San Francisco) and various commercial software packages. In the following sections, we describe the format of 3D datasets and several techniques that can be used to visualize the data.

Definitions and the User Interface

3D datasets. A data cube refers to any 3D (or higher) array of values—such as a stack of 2D black-and-white images representing a spatial volume. Unlike crystallographic and surface data, data cubes grow rapidly—a $1024 \times 1024 \times 1024$ data cube of 32-bit values occupies 4.29 gigabytes of memory. Each 3D element in the data cube is termed a voxel, analogous to a pixel in a 2D image. In tomography, each voxel in a data cube contains a value that represents intensity at a point (x,y,z) in the volume, which may relate to the composition or some other characteristic of the object at that point. For example, in medical X-ray computed tomography (CT), higher intensity represents denser material; a human head has brighter values at points where the hard skull is located than where soft brain matter is located [1,8]. In nanoscale STEM tomography, high-Z gold nanoparticles appear brighter than low-Z silica nanoparticles [4]. Because intensity values are provided at all voxels, not just at material surfaces, tomography reveals the entire 3D internal structure of an object [1].

3D data cubes are not limited to spatial volumes. In a hyperspectral map the third dimension is spectroscopic ($x-y$ -wavelength, or $x-y$ -energy) [9]. In a black-and-white movie, the third dimension is time ($x-y$ -time). In a tomographic tilt series, the data cube contains images at different projection

angles (*x-y*-angle). Although tomographic volume data is the most directly interpretable, *tomviz* is capable of visualizing any type of 3D data cube.

Loading volumetric data into *tomviz*. In *tomviz*, users can load a variety of datatypes stored as a TIFF, PNG, raw binary data, MRC, or HDF5 under “File > Open” in the top menubar. For convenience, *tomviz* comes packaged with an example dataset, a “Star Nanoparticle (Reconstruction),” so users can immediately explore different visualization techniques. The dataset contains the 3D structure of a Co₂P hyperbranched nanoparticle; it can be opened from the menubar under “Sample Data.” For further exploration of visualization techniques, five high-quality electron tomography datasets from the peer-reviewed literature have been made publicly available for researchers to use (Levin et. al., 2016). The datasets can be accessed under “Sample Datasets > Download More Datasets,” or they can be downloaded at <https://dx.doi.org/10.6084/m9.figshare.c.2185342>.

The *tomviz* user interface. New visualizations can be added from the *Visualization* menu or by clicking an icon from the Toolbar:



All *tomviz* visualizations are interactive 3D objects that can be rotated by clicking and dragging in the render window(s). The dataset and its associated visualizations—an *outline*, a

volume, and a *slice* in the case of Figure 1—are listed in the upper left-hand *Pipelines* (Figure 1a). As the user adds new visualization modules to a dataset, these will be displayed in the pipeline for that dataset. Each dataset that is opened in *tomviz* is assigned its own pipeline. This allows the user to easily keep track of all changes made to every dataset. Further properties for each selected visualization module or dataset are shown in the lower left-hand *Properties Panel* (Figure 1b).

In addition to a suite of visualization tools—the subject of this tutorial—3D data processing is also integrated into the *tomviz* user interface. A Python console (Figure 1c) can be opened from “Tools > Python” Console under the top menubar. Users can execute pre-written python scripts or create their own code using numpy, scipy, or ITK. The *Data Transform* and *Tomography* menus contain pre-written algorithms for image processing, alignment, and tomographic reconstruction of raw electron microscopy data. After these scripts are executed in Python, they are sent back to *tomviz* for immediate visualization. Data transforms appear in the pipeline in the order in which they are executed. Double-clicking on a transform listed in the pipeline opens a new window displaying the Python code used to implement the transformation, which the user may edit if desired. The *tomviz* pipeline uniquely preserves all steps for reproducible workflow.

After a dataset is loaded, a histogram panel appears (Figure 1d). Colors in the histogram represent a colormap that quantitatively correlates colors in a visualization to corresponding

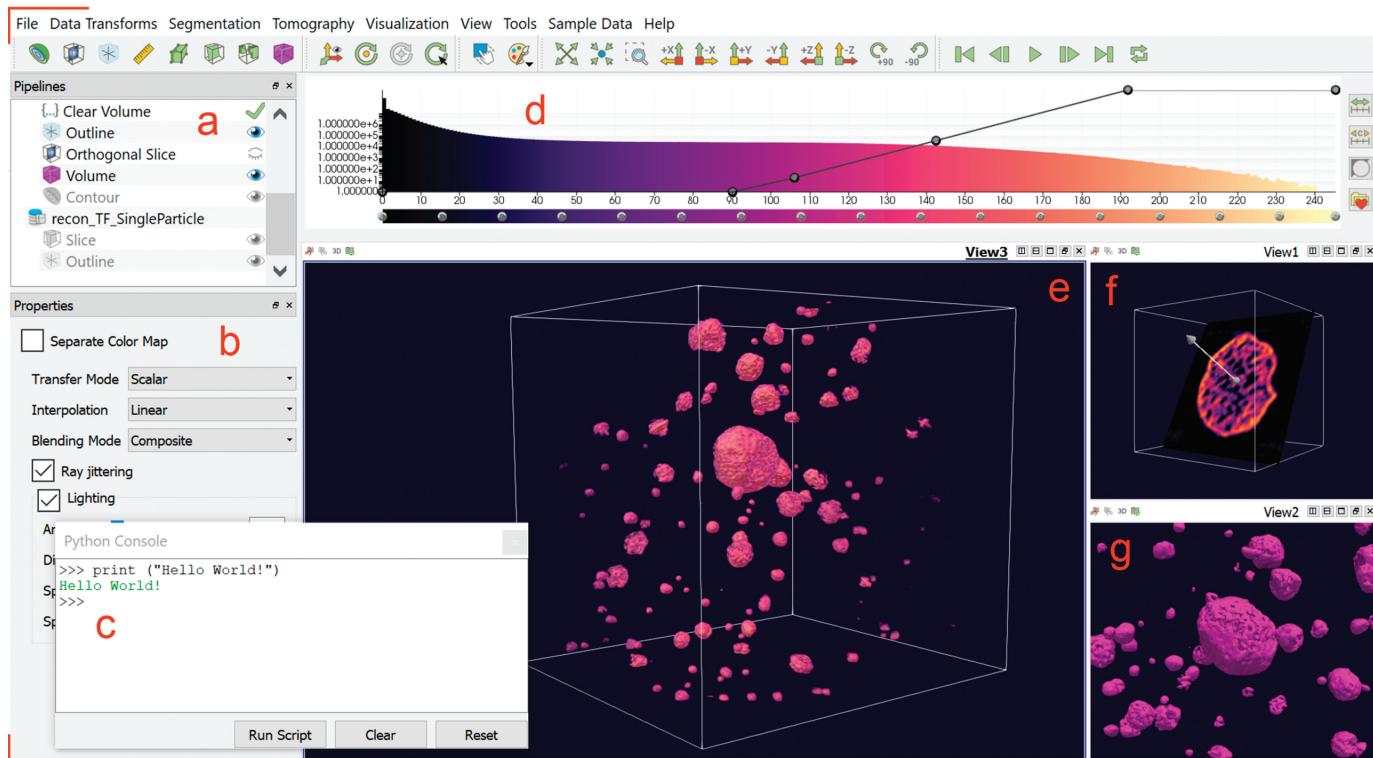


Figure 1: The *tomviz* graphical user interface for 3D visualization of tomographic data. Once a dataset has been loaded, the data pipeline is populated (a). A variety of visualization types are available to be used in combination for analysis—with parameters specified in the module properties panel (b). A Python interface offers advanced scientific processing of data (c). A histogram of voxel intensities is displayed top center (d). The line across the histogram represents the opacity map and can be altered interactively. Here a reconstruction of porous PtCu nanoparticles is visualized using a volume render (e), a non-orthogonal slice through one particle (f), and a contour surface (g). Tabs, and divisions within a tab, allow multiple simultaneous renderings and camera angles.

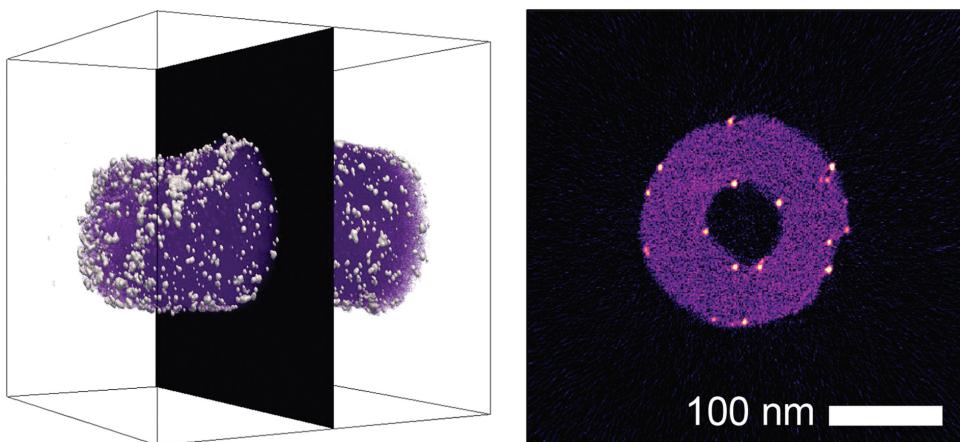


Figure 2: A volumetric reconstruction of platinum nanoparticles on a carbon nanofiber [10], showing the position of an extracted orthogonal slice. In the extracted slice, platinum nanoparticles (white/orange) can be identified on both the interior and exterior surfaces of the nanofiber (magenta).

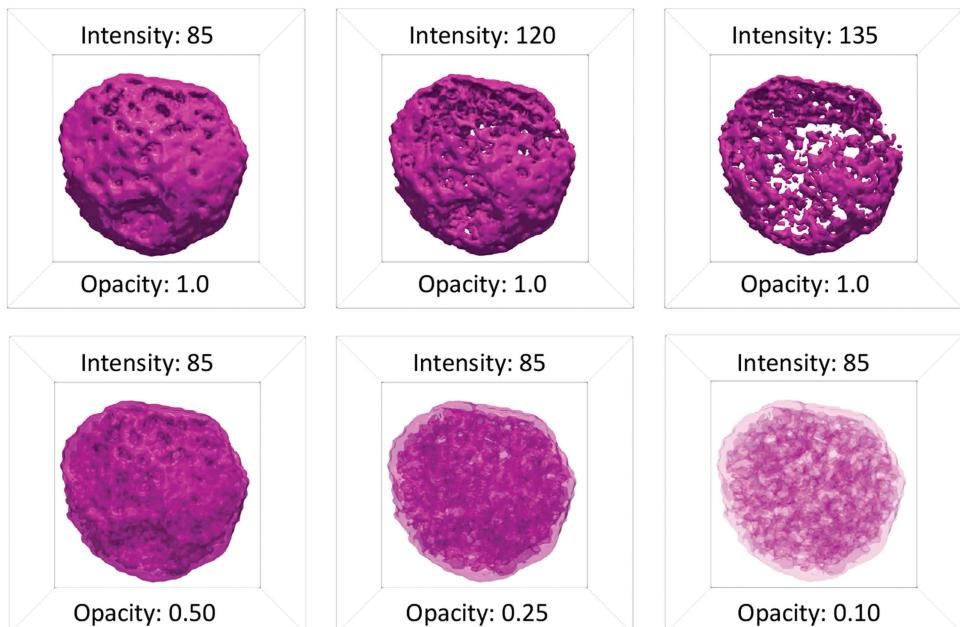


Figure 3: Surface renderings of a porous PtCu nanoparticle [10] at different values of intensity and surface opacity. At full opacity, only the exterior of a surface contour is visible. At low intensity, this exterior surface contour corresponds to the exterior surface of the nanoparticle. As the intensity of a surface contour is increased, features of the interior of the particle become visible. Varying the opacity at lower intensity allows the user to see through the exterior surface and observe the interior surfaces of the nebulous internal pore structure of the particle.

dataset values. This color map corresponds only to the dataset currently selected under *Pipelines*. Different color maps can be chosen from a list of presets, or the map may be adjusted interactively by clicking and dragging points on the color bar underneath the histogram.

By default, *tomviz* automatically adds a box outline and an orthogonal slice taken through the volume center in a *Render View* panel. Users may then add new visualizations of their own to the default *Render View* panel (Figure 1e). Additional *Render View* panels can be added from the Toolbar: allowing multiple visualizations to be shown simultaneously in separate panels, at different viewing angles (Figure 1f, Figure 1g).

3D Visualization Techniques for Volumetric Data

Three principal visualization methods are available in *tomviz*: slicing, surface rendering, and volume rendering. The visualization method used to produce the final 3D visualization can be highly variable depending on the dataset, the scientific questions to be answered, and aesthetic preference.

I. Slicing data with planes.

The simplest method for detailed examination of specific features of the internal 3D structure of an object is to inspect individual 2D sections—or slices—through the 3D dataset. The *tomviz* package features two different slice viewing tools. Orthogonal slicing allows users to view slices through the data perpendicular to one of the principal *x*, *y*, or *z* axes. Both the plane (*x*, *y*, or *z*), and the position of the slice can be altered interactively in the *Properties* panel (Figure 1b), as can the opacity/transparency of the slice. An example of orthogonal slicing is shown in Figure 2. In this figure, an orthogonal slice is extracted from the center of a reconstruction of platinum nanoparticles on a carbon nanofiber support. Platinum nanoparticles can be seen on both the interior and exterior surfaces of the nanofiber in this slice.

Non-orthogonal slicing offers greater flexibility, allowing users to view slices through the reconstruction at any orientation. The orientation and position of a non-orthogonal slice can be adjusted interactively by clicking and dragging the axis and slice in the *Render View* panel (Figure 1f) or manually by entering values in the *Properties* panel (Figure 1b).

II. Surface rendering. Surface rendering was originally applied to volumetric data to offer a more direct method of visualizing the 3D morphology of structures than 2D slices [11]. In surface rendering, a surface contour of constant intensity is generated from a user-specified intensity value using a fast-flying edges algorithm [12]. In *tomviz*, the intensity value of the surface contour can be entered manually or adjusted using a slider in the *Properties* panel (Figure 1b). The color, opacity/transparency, and lighting (specularity) of a surface render can also be altered in the *Properties* panel. Surfaces are rendered in a single color. By default, this color will change as the intensity of

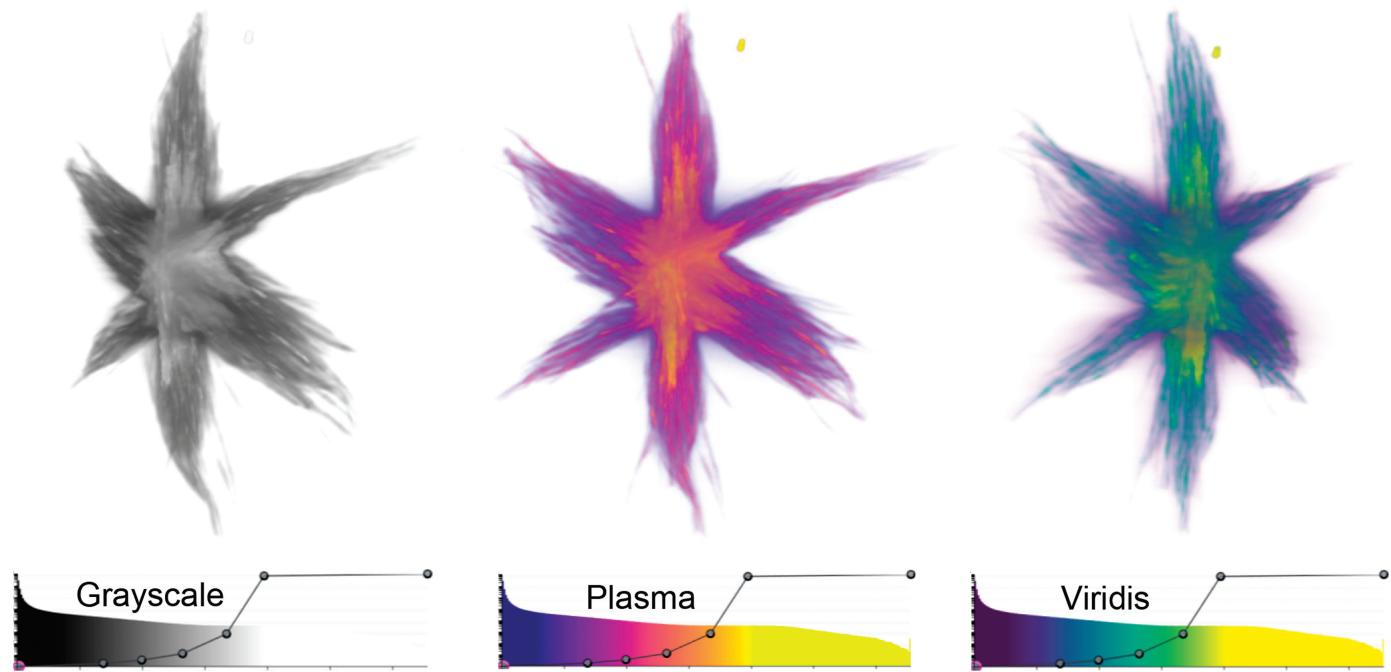


Figure 4: Volume renderings of a Co₂P nanoparticle [10] viewed from different perspectives using Grayscale, Plasma, and Viridis color-opacity maps. In these renderings, the dense internal core of the nanoparticle is displayed in lighter, more opaque colors, whilst the branches that protrude from the core are displayed as darker and more transparent colors. The color-opacity maps used in the visualizations are shown below each view of the particle.

the surface is changed, following the color map. Alternatively, the user can choose to use a fixed color for all surfaces. Figure 3 shows different isosurfaces of a porous PtCu nanoparticle with a fixed color, and different combinations of intensity and opacity. At full opacity, only the exterior of a surface can be visualized. Reducing the opacity of a surface contour allows the surface of interior structures, such as pores, to be visualized.

III. Volume rendering. Volume rendering was originally developed to visualize more complex 3D structures with many gray values, which are difficult to visualize by surface rendering [13]. In volume rendering, each voxel is assigned both a color and an opacity based on its intensity. The gray line overlaying the histogram of the data in the *tomviz* interface (Figure 1d) is the opacity map of the visualization, and together with the color map, this defines the color-opacity map. Figure 4 shows example volume renderings of a Co₂P star-shaped nanoparticle at different viewing angles, with their associated color-opacity maps. In *tomviz*, the color-opacity map can be adjusted interactively with dynamically updated 3D visualizations. The user may simply click and drag a point on the gray line overlaying the histogram to vary the opacity of different voxels in the volume render. In the examples shown in Figure 4, the color-opacity map has been set so that the opacity increases with increasing voxel intensity, and the color becomes increasingly light with increasing voxel intensity. The range of the color map has been adjusted to enhance the brightness and contrast of the visualization. Minor adjustments to the color-opacity map can have a noticeable influence on the final appearance of the visualization of an object. It is essential for users to adjust the color-opacity map to produce a volumetric visualization

that reflects the features of the sample that they have imaged. In addition, it is important to include the color-opacity map with any published scientific volumetric renderings to aid interpretation.

Bounding boxes and scale. A bounding box frame outlines the edges of a dataset, giving the viewer an impression of 3D depth perception with a non-orthogonal perspective on a 2D screen. In the same way that a parallel set of train tracks converges into the distance in the real world, so do most 3D visualizations—features in the foreground appear larger. A bounding box adds more than aesthetic appeal because it helps restore the interpretability of a 3D structure rendered on a 2D screen. Most of all, a bounding box allows scientists to put meaningful scale bars along an edge. In *tomviz*, voxel distances can be displayed by selecting “View > Show Axis Grid.” An example visualization with dimensional scales is shown in Figure 5.

Animation. In addition to 2D figures, an animation of a 3D visualization is an effective method of illustrating the 3D structure of an object. In *tomviz*, an animation tool allows users to produce simple animations, such as a 360° orbit around a 3D object, or more complex animations by moving the viewing angle between a series of user-specified positions. A still from an animation is shown in Figure 6. The full animation is available online in the digital edition of this issue at this web address: <https://doi.org/10.1017/S1551929517001213>. This animation shows a mixed volume and surface rendering of platinum nanoparticles decorating a carbon nanofiber of ~170 nm in diameter. The carbon fiber is visualized in black using a volume render, with the color-opacity map set such that only voxels

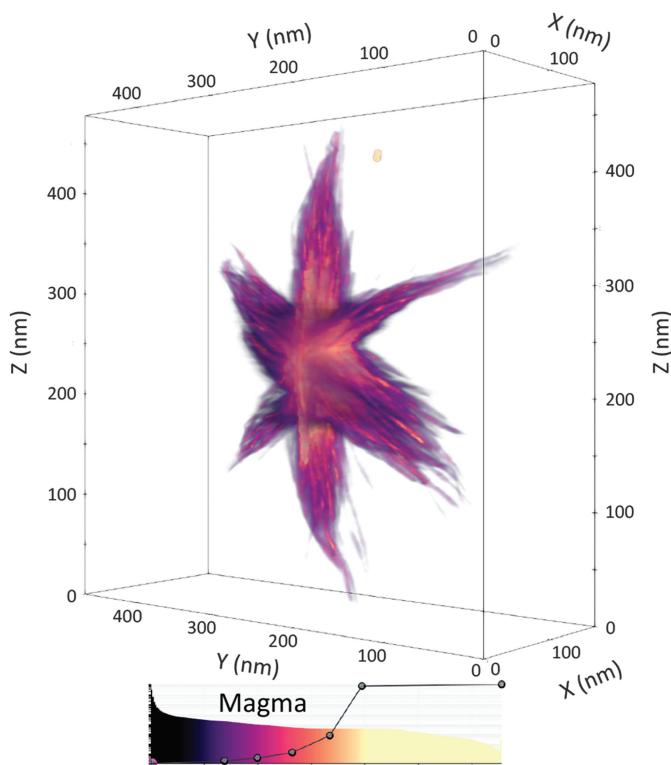


Figure 5: Volume rendering of the Co₂P nanoparticle from [Figure 4](#), enclosed in a bounding box with scale markers in nanometers shown on each axis of the box. Using perspective, the bounding box and scale together convey a sense of the 3D size of the particle. The color-opacity map of the volume render is shown underneath the visualization.

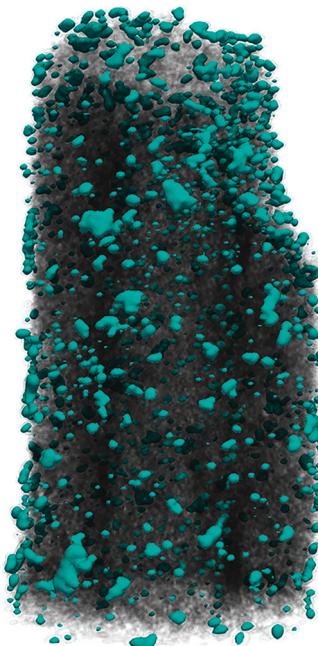


Figure 6: Still image from an animation of platinum particles (turquoise color) on a carbon fiber. The animation is available online in the digital edition of this issue at this web address: <https://doi.org/10.1017/S1551929517001213>.

with an intensity corresponding to the carbon fiber are visible. The platinum nanoparticles are visualized using a turquoise

surface contour, with the intensity of the contour set such that only platinum is visible.

Save and share. Whilst a high-quality manuscript figure can convey a sense of 3D structure, it is no substitute for interactive visualization and exploration of a volumetric dataset. With *tomviz*, the data, visualization pipeline, and software can all be shared with other scientists. Simply click “File > Save State” to share the *tomviz* state file (*.tvsm) and volumetric data with colleagues. For scientific publication, a screenshot of the visualization can be exported using the existing background or a transparent background by executing “File > Save Screenshot.”

Open-source license. *Tomviz* is made freely and publicly available as open-source software under the 3-clause BSD License that enables any party to copy, distribute, and make modifications to the software. This includes government organizations, for-profit and not-for-profit educational institutions, and commercial organizations, as well as individuals. *Tomviz* is hosted on Github and is open to further contributions from new users.

Conclusion

High-quality visualizations of 3D volumetric datasets are essential for interpreting and sharing tomographic data. In this tutorial, we have used the open-source *tomviz* software package to review the range in which volumetric data can be represented and explored—including 2D slices, surface contours, volume rendering, and combinations thereof. Ultimately, the final visualization will depend on the questions, goals, and aesthetic of the researcher. Readers are encouraged to explore these techniques further using their own datasets, or using publicly available STEM tomography datasets [10].

Acknowledgements

This work was supported by DOE Office of Science contract DE-SC0011385. The authors thank the many open-source developers and electron microscopists who have given valuable feedback and other contributions to the *tomviz* project. Elliot Padgett acknowledges support from NSF graduate research fellowship, DGE-1650441.

References

- [1] GN Hounsfield, *Computed Medical Imaging, Nobel Lectures, Physiology or Medicine 1971-1980*, ed. J Lindsten, World Scientific Publishing Co., Singapore, 1992.
- [2] DJ DeRosier and A Klug, *Nature* 217 (1968) 130–34.
- [3] RA Crowther et al., *Proc R Soc London, Ser A* 317 (1970) 319–40.
- [4] M Weyland et al., *J Phys Chem B* 105 (2001) 7882–86.
- [5] BJ Inkson et al., *Scr Mater* 45 (2001) 753–58.
- [6] A Cerezo et al., *Rev Sci Instrum* 59 (1988) 862.
- [7] A Cerezo et al., *Mater Today* 10(12) (2007) 36–42.
- [8] DH Phillips and JJ Lannutti, *NDT&E International* 30(6) (1997) 339–50.
- [9] P Cuevas et al., *Microsc Microanal* 18 (2012) 667–75.
- [10] BDA Levin et al., *Sci Data* 3 (2016) 160041.
- [11] WE Lorenson and HE Cline, *ACM SIGGRAPH Computer Graphics* 21(4) (1987) 163–9.
- [12] W Schroeder et al., *IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*, Chicago, 2015, p. 33–40.
- [13] A Kaufman et al., *Computer* 26(7) (1993) 51–64.