3D Visualization with ParaView

Random notes

Vedrana Andersen Dahl

December 9, 2019

Visualizing data in 3D often produces a comprehensive view of the imaged matter – and of the results obtained by analyzing volumes. While 3D visualization is possible using specialized libraries of general-purpose programming languages as Python, or using built-in functionalities MATLAB , specialized software that allows better interactivity is sometimes a faster way of producing visually pleasing images.

Here I provide some guidelines for using ParaView `https://www.paraview.org` for 3D visualization of volumetric data and analysis. Guidelines are written on an *on-demand* basis. The focus is on a few specific analysis pipelines, including segmentation, orientation analysis, connected components analysis and analyzing local thickness. For an overview of the vast visualization possibilities provided by ParaView look at ParaView Guide at `https://www.paraview.org/paraview-guide`.

ParaView is cool because it is free and open-source. It runs on Windows, Mac OS X, Linux and a number of other platforms. It is developed for handling large datasets, making it suitable for X-ray CT data. It can be used interactively, but it also offers scripting support through Python – here I cover only interactive use of ParaView.

On the downside, due to the large amount of options available in ParaView, it is sometimes difficult to find the desired functionality. And experimenting is may be challenging, since unsuitable use of ParaView might get it to hang or crash, especially when working on larger datasets. It is always a good idea to experiment on smaller data. Another source of frustration is a user interface, which requires practice. I've spent many hours trying to figure out functionality which I believed was faulty, to learn that I need to press enter after inputting parameters.

Visualizing 3D gray-scale data, for example X-ray CT data, is a first ting to try. Depending on your set-up, a typical workflow using ParaView might be:

- Start ParaView.
- Open the data (in the File menu choose Open). For the data to load, you may need to click Apply in the Properties window (if the Properties window not visible, check its visibility in the View menu).
- Turn on the visualization of the data by clicking the small eye icon next to the filename in the Pipeline Browser window (if the Pipeline Browser window is not visible, check its visibility in View menu). Visualization probably turned on when opening and loading the data, so this step may not be needed.
- Choose Representation as Volume by selecting the Volume from the unnamed drop-down menu which has the default value set to Outline.
- Adjust the Mapping Data by dragging dragging the mapping curve which defines the transparency associated with each gray value. This curve is shown in the Color Map Editor (if Color Map Editor is not visible, check its visibility in View menu).
- You may want to change the colormap which defines a color associated with each gray value. The selection of useful colormaps is available in the Color Map Editor when clicking the folder+heart icon left to the Mapping Data, and even more colormaps are shown if you open additional options by clicking a small gear icon.
- You may want to visualize an isosurface produced from your data. For this, click a contour icon which looks like half of an onion. Remember to adjust the isovalue under properties of the contour. Also remember to turn on the visualization, with representation set to Surface.
- Adjust other settings like background color (Palette icon), visualization of orientation axes (checkbox at the bottom of Properties window) and show/hide of color legend (rainbow icon at the top of Color Map Edditor).

Slightly more advanced visualization possibilities include:

- Clip to produce a clipped visualization of data by adding a Clip filter. Beware, this produces a copy of the data in the unstructured mesh format. This is very costly for volumetric data. For volumes, see whether you can achieve the desired effect by using Extract Subset instead.
- Add Light in Light Inspector window (access Light Inspector window from by checking its visibility in View menu)
- Animate the visualization in Animation View window (access it in View menu)
- Select from standard viewpoints Adjust Camera by selecting a tiny camera-like icon just over rendered visualization.
- Manage Viewpoints (add custom Viewpoint, import and export Viewpoints) via larger icons showing camera with wrench and plus symbol.

Combining different data is especially useful when visualizing the results of the analysis on top of the data. The visualization from different sources or steps can an be turned on or of in the Pipeline Browser. Examples of combining different data are shown further down in gallery.

Using colors to to visualize the outcome of data analysis is often advantageous, as for example when showing connected component analysis, or orientation analysis. Colors in ParaView are normally computed by scalar mapping: the active colormap defines mapping of the scalar values from the data to the RGBA colors, while transfer function defines transparency, and both are shown in Mapping Data part of Color Map Editor. Another option of displaying analysis in colors is to save the desired RGBA colors with the data and turn off the default scalar mapping. You can do this if you can find well-hidden functionality.

Under Properties, open additional options by clicking a small gear icon and uncheck Map Scalars. This is a crucial step for making ParaView show the RGBA color saved in the file, and not mapping it (or rather, either its magnitude, or one of the channels) using the active colormap. In my version of ParaView, even with Map Scalars turned off, the color legend will still show, misleadingly, the mapped colors. So it is advisable to turn color legend off by clicking a rainbow icon in Color Map Editor. Furthermore, the transparency control trough dragging the transfer function will still be working – which is nice, but a bit confusing since transfer function visible in Mapping Data is still shown with the (not active) colormap.

A combination of a surface (contour) and a color information is my favorite visualization type. I achieve by loading two volumes: a grayscale volume which is to define a surface, and a RGBA volume which is to give a color to surface. The two volumes can be combined by choosing an Append Attributes filter with both volumes selected. The resulting volume can then be further processed using a Contour filter. The filter will automatically create contour by gray values, but you might need to adjust the isovalue. Coloring will automatically be set to RGBA values, but will probably use mapping of color magnitute, so you should disable Map Scalars.

PraView state file (.pvsm file) is extremely useful for saving the visualization settings and sharing the results. To save the settings, chose Save State in File menu and choose a name for the state file. The state file has an extension .pvsm, but is actually a tagged xml file. Some of the settings, like background color or viewpoints, are not saved in the state file. The state will be saved with the *absolute paths* to the data sources. When using Load State in File menu, the default is to Use File Names From State, which will load data from sources used when saving the state.

When ParaView state file is provided with the data, for example when sharing the data between users, the usual workflow is:

- Start ParaView.
- Load State in File menu.
- Choose Search files under specified directory and point Data Directory to where the data is stored.

Gallery Visualization showing 3D data and the results of volumetric analysis are shown in Figures at the end of this document.

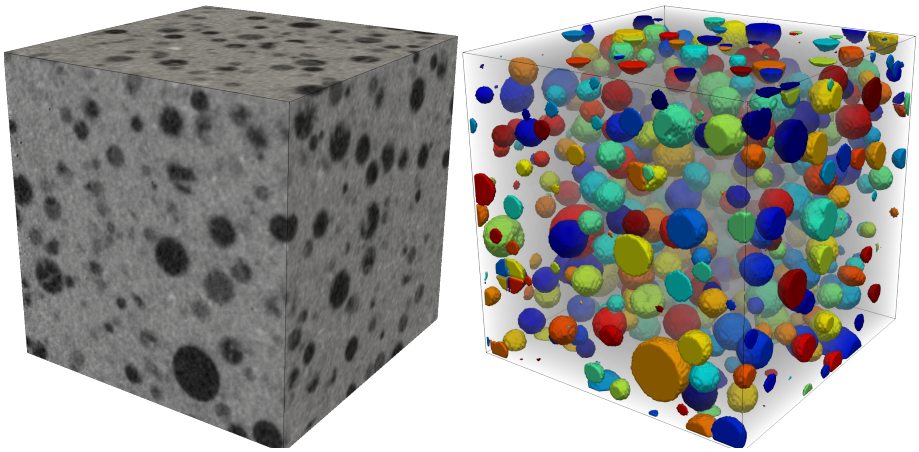

Figure 1: Detecting porosity in volumetric data of cement samples. On left, surface visualization, on right combination of grayscale volume rendering and surface rendering of pores which utilizes a colored label volume. The colors are used to distinguish individual pore, and label volume for assigning color is created outside ParaView.

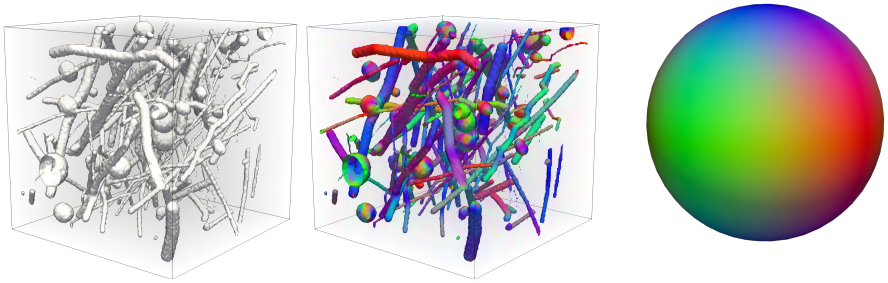

Figure 2: Visualizing the orientation of fibers using colors. Red color indicates a fibre in x direction, green is used for y direction, and blue for z direction – also visible in the spherical colormap shown in rightmost image. The orientation of fibre segments is calculated outside ParaView, and saved as a volumetric RGBA image.

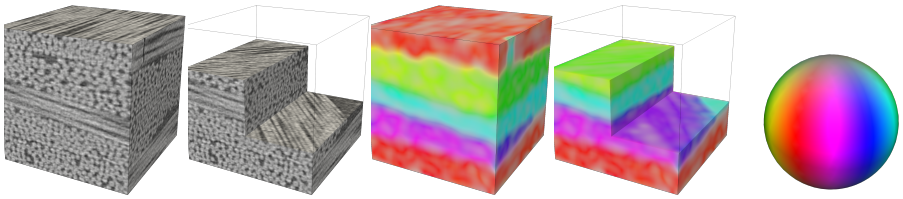

Figure 3: Using color to visualize the orientation of fiber bundles in composite material. Orientations are computed outside PareView, and saved as a volumetric RGBA image. Colors are assigned to voxels according to the azimuth of the local orientation, which is suitable for this case since all bundle orientations lie in xy plane.

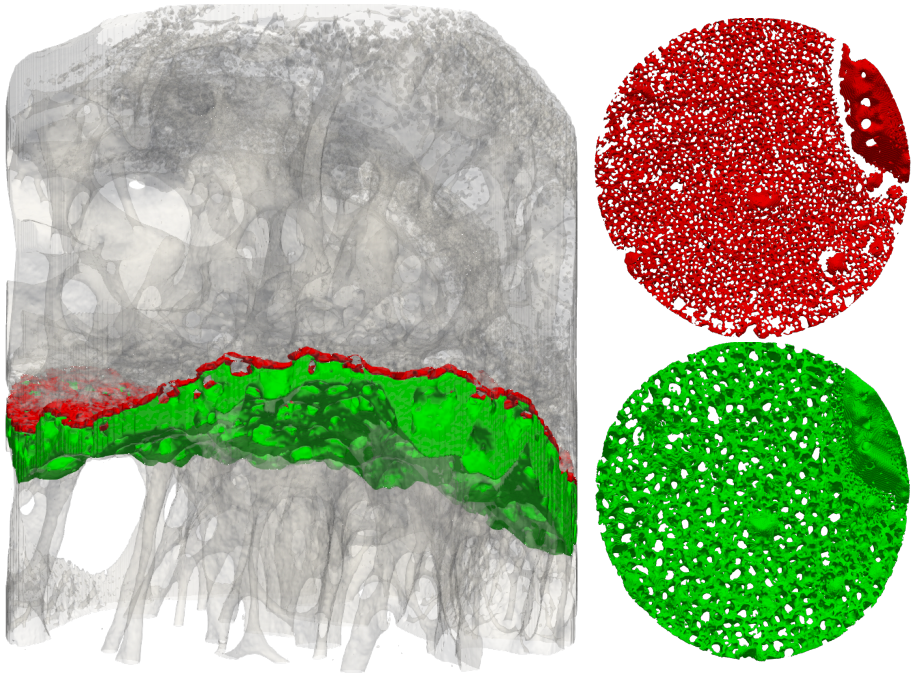

Figure 4: Detecting growth plane in mice tibia to quantify trabecular structure in different layers of growth plane. Combination of volume rendering and surface visualization. Surfaces are created using isosurface filter on small subvolumes (layers) detected outside PareView.

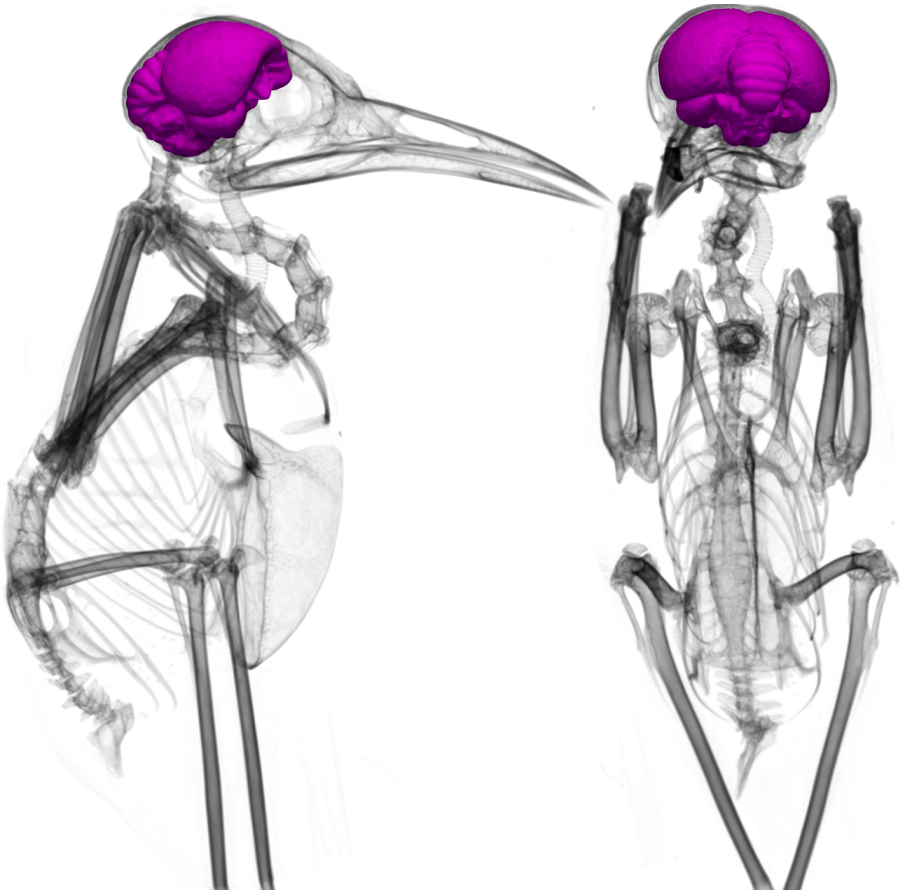

Figure 5: Detecting bird brain to measure brain volume. Combination of volume rendering (gray) and visualizing a mesh created outside ParaView (purple).

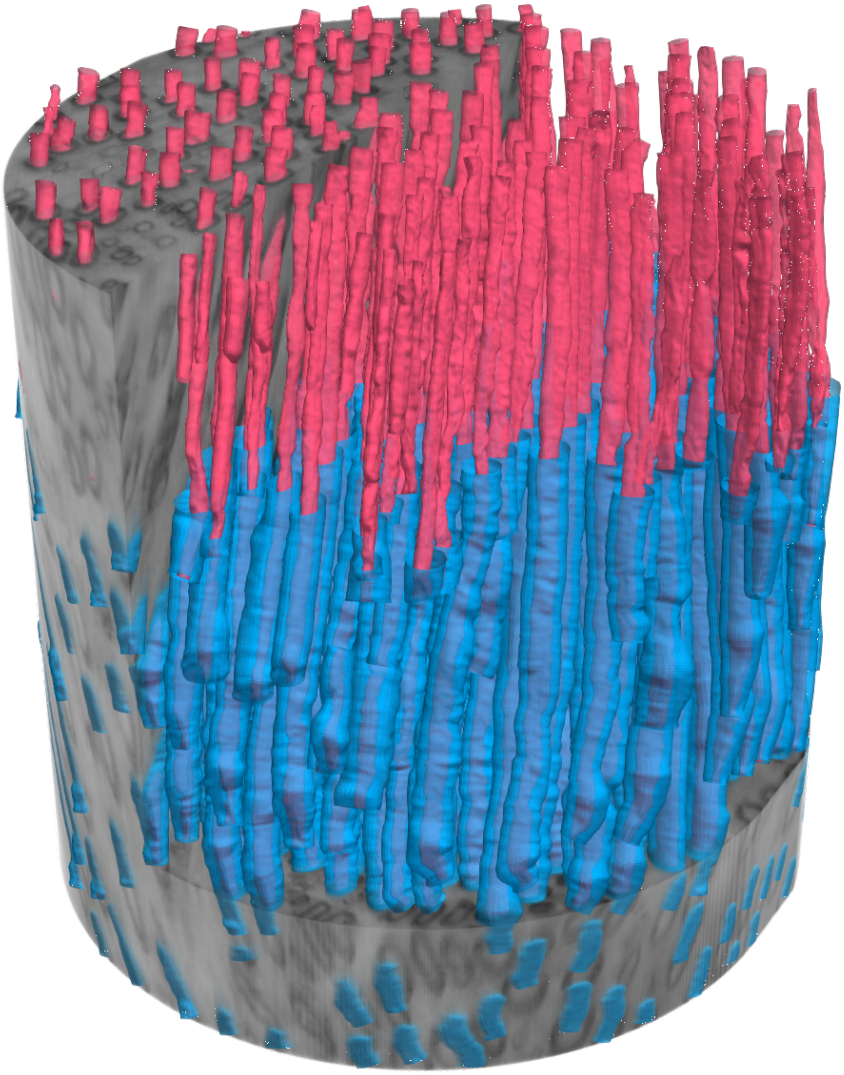

Figure 6: Detecting nerves in the volume depicting a peripheral nerve. Combination of volume rendering (gray) and visualizing meshes created outside ParaView (red and blue).

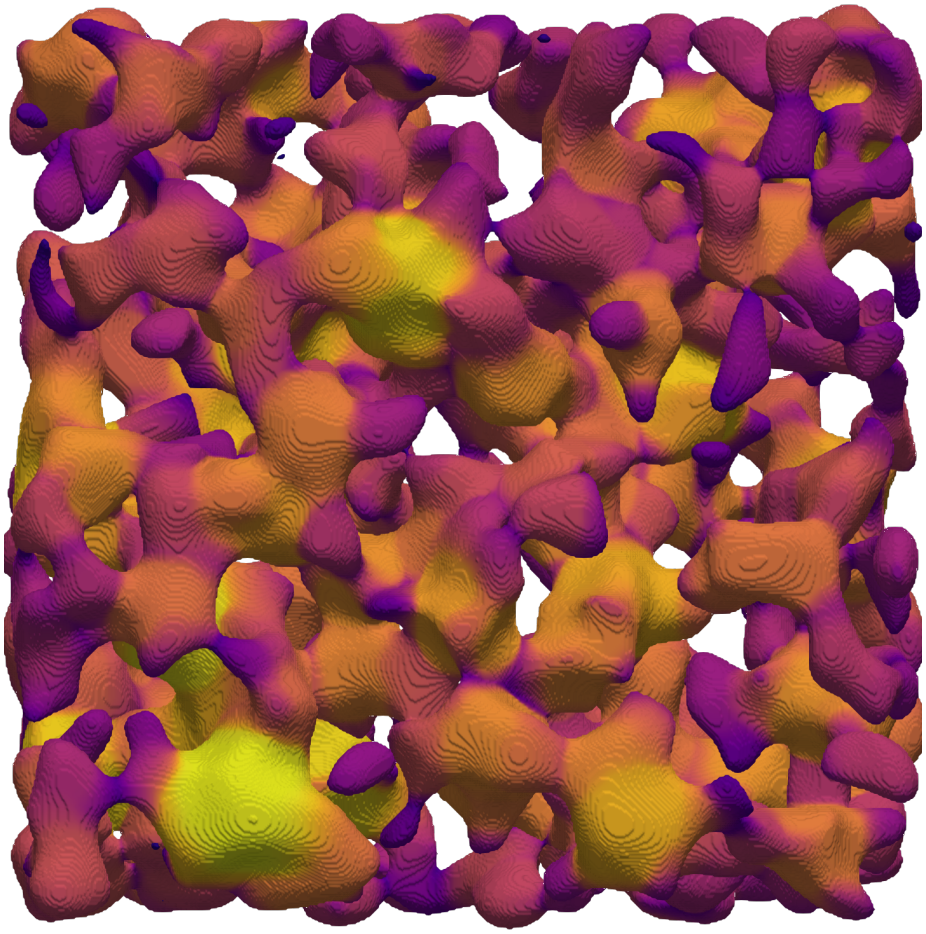

Figure 7: Local thickness of a structure visualized using a colored surface mesh. Local thickness is computed outside ParaView and saved using a volumetric RGBA image.