1370
doi:10.1017/S143192761900758X
Microsc. Microanal. 25 (Suppl 2), 2019

Scalable Imaging Science Tools to Support Increasingly Efficient Workflows for Research Tasks on Massive Images

Christian Goetze[1*], Christopher T. Zugates[2], Thomas Ruth[1], Paul Boenisch[1], and Adam Brady-Myerov[2]

[1.] arivis AG, Rostock, Germany
[2.] arivis inc., Washington, DC, USA
* Corresponding author: christian.goetze@arivis.com

Several years ago, an experiment would produce a single dataset with megabytes of images. These images were relatively easy to manage; manual analysis was clearly possible. Today, driven by new imaging devices and ambitious scientific questions, experiments yield Terabytes (TB) of high dimensional image data, presenting huge challenges to handling and management. New approaches are needed to extract and share information from these data. We recently collaborated with researchers who made the first biological application of Expansion Lattice Light Sheet Microscopy (ExLLSM) [1]. They made millions of single images, forming thousands of single stacks, which they then stitched together into massive volumes exceeding 10 TB per dataset. For several crucial parts of the research process, we helped with examination, structure masking/analysis, and production of sharable images and movies. As we worked to develop and optimize workflows, we were inspired to new ideas, concepts, and tools that we are keen to share here.

The huge ExLLSM images were a formidable challenge to our current storage and visualization platform. For downstream full-resolution analysis on cluster systems, it was necessary to efficiently explore these massive image volumes in slice-by-slice and 3D views, and structures-of-interest had to be efficiently masked and subsequently examined at the full resolution of the images. Furthermore, we needed to refine the masks then extract the voxels therein for measurements.

Initially, we made masks by drawing and interpolating between polygons on 2D slices of the image at its full resolution. Although this workflow could proceed continuously, some structures were ambiguous in 2D view, requiring occasional 3D rendering of the area at full resolution. This inspired us to try creating mask segments in our VR solution. We found using a Virtual Environment makes the process faster and delivers a more reliable result, as complicated neuron branches, crossings, and contacts are seen directly and masked naturally in 3D at the time of perception. These VR-derived masks were given to filters and mathematical segment generation routines to segment neurons and to create other useful derivatives of the mask (e.g. masks dilated from the neuron surface by specified distances). We could leverage the multi-resolution format of the raw pixel data to give lower zoom levels to mathematical operations so that refinement of the masks could proceed efficiently.

As ExLLSM provides a substantial increase in data resolution, it was crucial to show the original data resolution in videos. Due to the data size, however, any rendering close to or at original resolution requires an out-of-core approach, both in terms of GPU and main memory. A first attempt was made using our conventional renderer supported by 512GB of RAM. However, it was too slow as huge amounts of data had to be loaded from the hard drive for each movie frame. To reduce the memory footprint, a second attempt was made using a Dynamic Level of Detail (DLoD) approach. Data were split into overlapping 3D blocks with each block loading just the resolution to satisfy a screen error metric based on viewer distance. As the data needed is reduced, DLoD rendering increased the speed by 2-10x, depending on scene content. To achieve the desired data quality, however, it still required significant data reloads from the hard disk for every frame. Another downside were “pop-up artifacts” along the camera path. As the distance of a block changes, a new data resolution might be required, leading to noticeable changes in the data due to down sampling interpolation.

To avoid these effects, changes of resolutions have to be blended while rendering, significantly increasing memory footprint and thus slowing down rendering.

To achieve the desired speed and quality, we finally developed a new rendering technique combining scene-analysis, tiled rendering, and LoD to use localized in-core rendering even with original data resolution. We tiled the output image into a number of 2D patches (e.g. 12x8) and then ran the entire movie export pipeline for each individual patch, ensuring the required data for a patch throughout the movie was small enough to fit into main memory and, over short sections, onto the GPU to make use of temporal coherence in caching. Additionally, the movie rendering was sectioned into areas with similar data resolution, allowing us to render each section individually (with overlap to adjacent sections) and then blend the transitions afterwards. This allowed for resolution changes with no pop-ups and without requiring additional memory during rendering. Finally, the individually rendered and blended patches were stitched together to create full movie frames which were then encoded into the final movie without further post-processing. Compared to conventional rendering, speed improvements ranged from 10- to 100-fold allowing us to present the data in original resolution within a reasonable amount of time.

We think, in future we can optimize the new rendering process as it may be possible to minimize the amount of data to read by analyzing which portion of data is needed for which patch in which frame. Data portions needed in multiple patch/frame combinations could then be re-used by optimizing the rendering order.

References:

[1] R. Gao et al., Science **363** (2019), p. 6424.

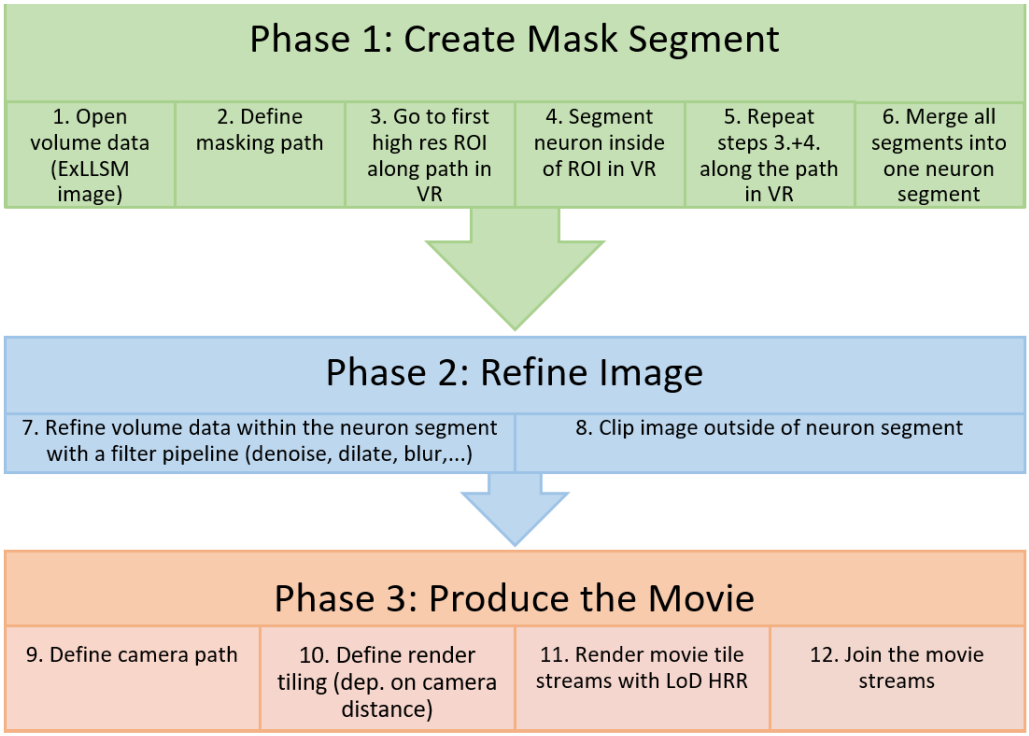

Figure 1. Overall workflow for working with multi-TB image volumes that we describe above.

https://doi.org/10.1017/S143192761900758X Published online by Cambridge University Press