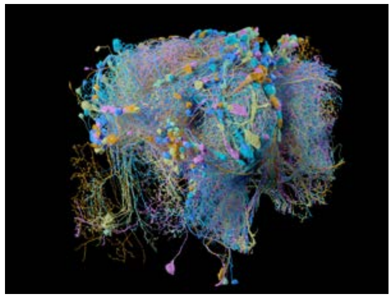
Open Science Software Initiative Proposal: Cell Videos from High-Level Descriptions

Philip Hubbard, Scientific Computing Software
November 2021



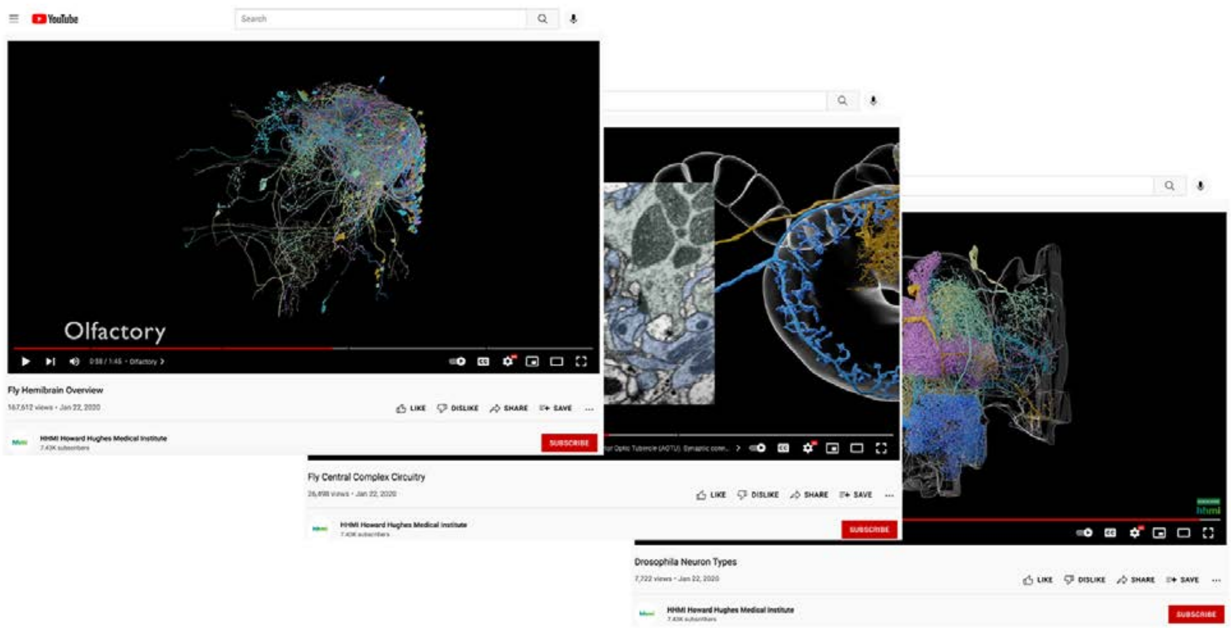
Introduction

The FlyEM project used the neuVid system (https://github.com/connectome-neuprint/neuVid) to create videos that increased the impact of publications and other publicity. This proposal outlines how that approach could be extended in a new system, CellVid, that could benefit projects within Janelia, like COSEM/CellMap, as well as external projects.

The neuVid system generates anatomical videos from high-level descriptions using the Python API of Blender (https://www.blender.org). The descriptions involve statements like "frame the camera on this group of neurons over the next 5 seconds" (in slightly more structured form), leaving the system to compute the geometric details and smooth transitions; the system outputs a file that Blender then processes to produce the frames of the animation. High-level descriptions enable rapid iteration to reach a desired animation, and simple reuse of ideas from previous animations, thus improving productivity. Another animation system using high-level descriptions is 3Dscript (https://imagej.net/plugins/3dscript), but it lacks the integration with Blender that is a strength of neuVid.

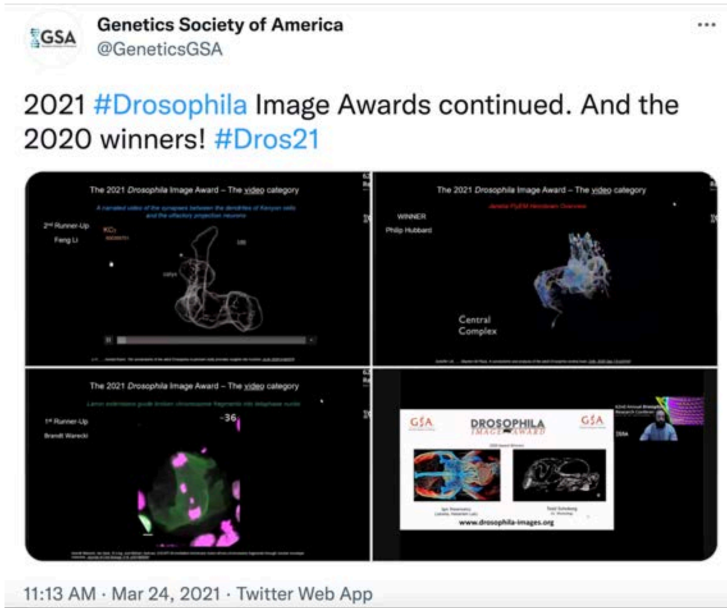
A guiding principle of neuVid is the use of sensible defaults. The goal is to make the system inviting to users by producing an acceptable output animation from even a minimal input description, with more detailed input increasing the sophistication of the result. For example, the input "frame the camera on this group of neurons" orients the camera so it views the fly face on, which is a standard orientation expected by scientists in the field, with the camera positioned so the group of neurons mostly fill the view; note that this useful result requires no mention of specific spatial coordinates. As another example, an input not mentioning colors prompts the system to choose neuron colors automatically, cycling through a palette that tries to avoid problems with color-vision deficiency while maintaining perceptual uniformity. The defaults and conventions were developed in consultation with a FlyEM scientist, Shin-ya Takemura. As a validation of the approach, a Janelia CAT member not involved in the system development, Marisa Dreher, used the system to produce dozens of videos for publication [Feng Li et al, 2020].

The neuVid system has contributed to the visibility and impact of the FlyEM project. The first use of neuVid was three videos for the FlyEM hemibrain dataset release in January 2020. These videos have been viewed more than 200,000 times on YouTube.



Images from these videos have been republished widely, including in *National Geographic* and *Nature.*

Dozens of additional videos produced with neuVid have appeared in Janelia publications [Brad K Hulse, et al., 2020][Feng Li, et al., 2020][Louis K Scheffer, et al., 2020]. A neuVid video won the 2021 Drosophila Image Award from the Genetics Society of America, and another neuVid video was second runner up.



The goal of CellVid is to support similar visibility for other projects, both internal and external to Janelia.

Proposed work

The new CellVid system will have its own Github repository, starting as a copy of the original neuVid repo. The following list of tasks to develop CellVid is presented in roughly the order it would make sense to do the work. Finishing the whole list would be the ideal outcome, but each completed task would provide incremental benefit, so the project can be scaled to the available funding. The work requires experience with 3D graphics in general and Blender in particular, and I will be happy to commit to doing the work myself.

Port to Blender 2.80+

The Blender Python API has changed slightly since version 2.79, the version for which neuVid was developed. The CellVid scripts will be updated to work with the latest versions of Blender.

Impact: The existing neuVid functionality will be easier to use for more people because it will no longer require a special installation of an obsolete version of Blender. This upgrade also prepares for the adoption of functional improvements from newer versions of Blender (e.g., the improved Cycles renderer; see below).

Time estimate: 1 FTE week.

Experiment with COSEM/CellMap data

Rendering FlyEM videos involved handling large amounts of polygonal data from neuron segmentations, but the full datasets from COSEM/CellMap tend to be even larger. Experiments will be necessary to determine what kinds of polygonal data from organelle segmentations can be handled with current techniques, and where new work will be needed (e.g., direct volume rendering, dynamic levels of detail; see below).

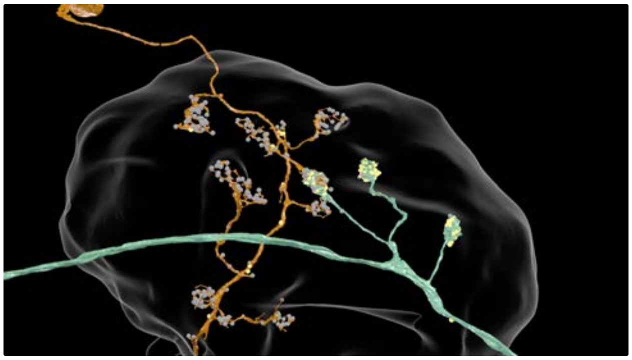
Impact: Seeing videos of the COSEM/CellMap data will be interesting, even if only partial datasets can be supported with current techniques. This work also will be the first example of using the system with data other than the original FlyEM data for which it was designed.

Time estimate: 2 FTE weeks.

Extend the description language

The commands in the high-level description language supported by neuVid will continue to be useful in CellVid, but additional commands will help with the needs of projects like COSEM/CellMap:

- Simple camera placement options, like "from back-left", will extend the fly-specific conventions currently supported, while maintaining the goal of avoiding the need for specific spatial coordinates.
- More flexible options for choosing objects' overall appearance will extend the current conventions (e.g., using silhouettes only for ROIs, or regions of interest).



Impact: Users will have more ways to tune their videos while keeping the benefits of sensible defaults.

Time estimate: 1 FTE week.

Support the Cycles renderer

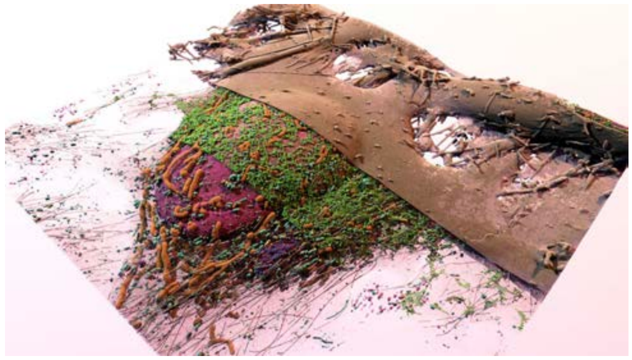
Blender supports multiple renderers, and neuVid works with the basic Blender "scanline" renderer, plus the Octane "path-tracing" renderer (https://home.otoy.com/render/octane-render). Octane efficiently generates realistic shading and shadows using GPUs but requires a paid commercial license. Blender also supports a free path-tracing renderer, Cycles, and adopting it will require some work to generate its specifications for visual parameters (i.e., materials). It will be interesting to see how the performance and quality produced by CellVid using Cycles compares with Octane.

Impact: Supporting Cycles will improve the quality that can be produced for free with no commercial license. Also, it is possible that Cycles is better than Octane at falling back to using the CPU when the GPUs reach capacity, so it may be able to support larger polygonal datasets.

Time estimate: 1 FTE week.

Support direct volume rendering

Another approach that may help with larger datasets is to abandon the use of polygonal data altogether, with rendering processing the voxel data directly. This approach, known as direct volume rendering, is how Stephan Saalfeld produced some still images of COSEM data. The implementation he used is the POV-Ray system (http://www.povray.org), but both Octane and Cycles support direct volume rendering.



In those systems the typical application is rendering fire or smoke for cinematic purposes, but scientific visualization should be possible. Voxel data can be represented with OpenVDB (https://www.openvdb.org), which is a standard and works with GPUs via the NanoVDB package.

Impact: Support for direct volume rendering will expand the visual effects CellVid can produce, with different effects on memory usage and performance that are worth exploring. Rendering directly from the voxels simplifies the workflow, in that there is no need to generate polygonal meshes or LODs (levels of detail) thereof from the segmentation. OpenVDB advertises support for time-varying data, which fits with another proposed topic for work (see below).

Time estimate: 3 FTE weeks.

Support cut-out volumes

Stephan Saalfeld's image (above) demonstrates the appeal of revealing hidden structures by cutting out the other structures doing the hiding. When the high-level description language is expanded to allow the specification of animated cut-out volumes, the choices for implementation include Boolean operations in Blender and clipping materials in Octane.

Impact: Revealing hidden structures with animated cut-out volumes can look better than the use of animated transparency in some cases and will expand the creative choices available to video authors.

Time estimate: 3 FTE weeks.

Support chunked meshes for levels of detail (LODs)

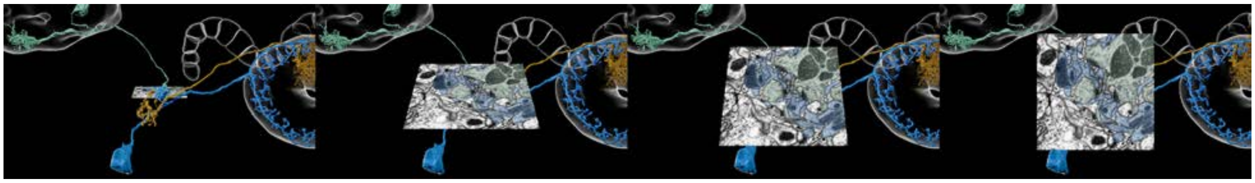
A way to handle large polygonal datasets is to use LODs, where the detail is higher closer to the camera and lower further away (where it is less visible). To use this technique effectively on a large object that may have both closer and further parts, the object's polygonal mesh mush be chunked, or segmented spatially so different parts can use different LODs. Chunking is computed already for COSEM data, because the Neuroglancer renderer used in OpenOrganelle (https://openorganelle.janelia.org) can exploit it. Octane and Cycles do not support chunking directly, and CellVid will need to explicitly determine the appropriate LOD for each chunk before passing it to the renderer for the current frame. This processing will fit into the `render.py` script carried over from neuVid, which already has the capability of special calculations before the rendering of each frame.

Impact: This dynamic approach to LODs will be more convenient and effective than the static, manual approach that is necessary with neuVid.

Time estimate: 3 FTE weeks.

Improve the workflow for grayscale

Some videos include grayscale images from the original EM data to show the nuances of post-synaptic densities [Feng Li, et al., 2020].



Currently, the workflow for adding these images is awkward, involving several manual steps in FlyEM's NeuTu proofreading application. Installation of NeuTu is nontrivial and should be unnecessary, since NeuTu is being used primarily as a means of connecting to the underlying image storage in DVID. For FlyEM data, an improvement will be Python scripts that access DVID directly, to fetch the grayscale images and add false coloring for the segmented bodies being emphasized. For other projects like COSEM/CellMap, the solution will depend on the particular details of how grayscale data is served, and also on what ways of showing the grayscale data in videos seem most appealing.

Impact: A simpler workflow will encourage authors to add grayscale when it makes sense for the narrative of a video.

Time estimate: 3 FTE weeks.

Document advanced features

There is considerable tutorial documentation for neuVid (https://github.com/connectome-neuprint/neuVid/tree/master/documentation), but for CellVid it needs to be extended to include the details of more advanced operations (e.g., adding grayscale, or using the most sophisticated rendering options).

Impact: Better, more complete documentation will allow more scientists to use the system successfully, especially those not onsite at Janelia.

Time estimate: 1 FTE week.

Produce "making of" tutorials for outreach

When Marisa Dreher learned neuVid to make videos for publication [Feng Li et al, 2020], what added to the effectiveness of the Github documentation was an in-person session going over how to write high-level descriptions for the planned videos. Similar "making of" presentations covering specific FlyEM videos on YouTube or in publications will help Janelia users, and recordings of such presentations posted on YouTube will help external users. For further educational effect, we will share the actual text of the high-level descriptions for the videos described.

Impact: Results-oriented training will invite new users to try the system and will give useful tips to more experienced users.

Time estimate: 3 FTE weeks.

Add unit tests

The neuVid scripts contain a few preparatory calculations that lend themselves to unit tests. An example is some analysis of time intervals, to determine which polygonal meshes are visible at which times, to reduce the data that must be sent to the Octane renderer (which runs as a separate process and operates more efficiently when only the necessary meshes are sent).

Impact: Unit tests will raise confidence in the system and make it easier to avoid problems when incorporating changes to the code, from either the original developers or external contributors.

Time estimate: 1 FTE week.

Support the PBRT renderer

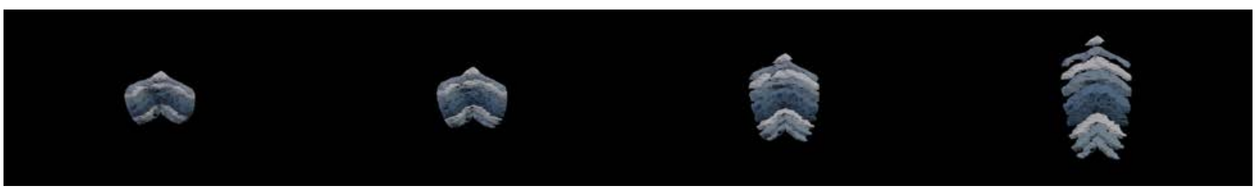
PBRT (https://github.com/mmp/pbrt-v3) is a highly regarded path-tracing renderer known to be able to handle very large amounts of polygonal data. Its next release will include performance enhancements like better GPU support (https://github.com/mmp/pbrt-v4). Supporting it will require some effort, as it does not directly integrate with Blender.

Impact: PBRT is likely to be able to handle datasets too large for other renderers, making it a least a useful option of last resort.

Time estimate: 3 FTE weeks.

Support animated parts "explosion"

A sophisticated way to reveal hidden structures is to show the "explosion" or moving aside of the hiding structures. The implementation of this feature will be based on an experimental, unpolished change to neuVid, used to animate the separation of layers of an ROI for a publication video [Feng Li, et al., 2020].



Impact: Some scientific visualizations use explosion to great effect, and it will be a nice addition to the repertoire of effects available in CellVid.

Time estimate: 1 FTE week.

Support 4D data

Janelia's research area two, 4DCP, will explicitly deal with time-varying phenomena. Some technologies proposed to be added to CellVid, like OpenVDB (see above), include at least some level of support for 4D data. The best way to fully integrate 4D data into CellVid will require investigation.

Impact: EM-based projects like COSEM/CellMap will not need 4D support any time soon, but other 4DCP projects will, and might be able to simplify their video-generation needs with CellVid given this capability.

Time estimate: 7 FTE weeks.

Acknowledgements

Konrad Rokicki, Stephan Saalfeld and Aubrey Weigel contributed valuable commentary on this proposal.

References

Brad K Hulse, et al., "A connectome of the Drosophila central complex reveals network motifs suitable for flexible navigation and context-dependent action selection," *eLife*, 2020, https://doi.org/10.7554/eLife.66039.

Feng Li, et al., "The connectome of the adult Drosophila mushroom body provides insights into function," *eLife*, 2020, https://doi.org/10.7554/eLife.62576.

Louis K Scheffer, et al., "A connectome and analysis of the adult Drosophila central brain," *eLife*, 2020, https://doi.org/10.7554/eLife.57443.