Informatics

Visualization and Interaction Techniques for the Exploration of the Fruit Fly's Neural Structure

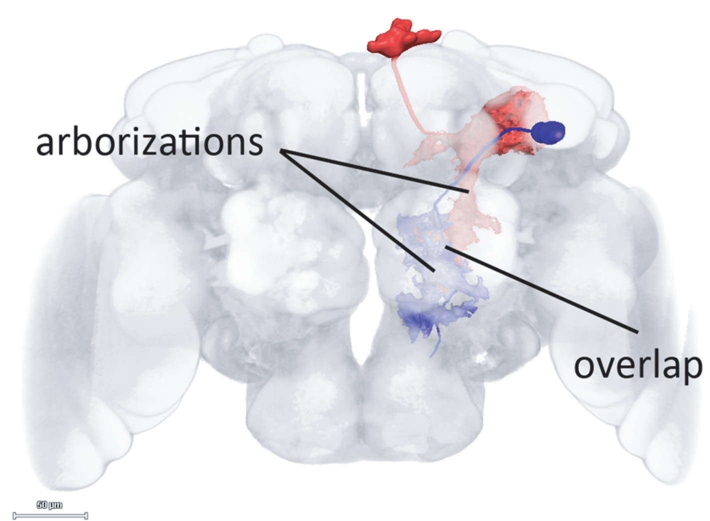
Nicolas Swoboda
Visual Computing

TU Wien Informatics
Institute of Visual Computing & Human-Centered Technology
Research Unit of Computer Graphics
Supervisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistant Supervisor: Dipl.-Math. Dr.techn. Katja Bühler
Contact: swoboda@vrvis.at, buehler@vrvis.at

Problem Statement

We want to support neurobiologists who link complex behaviour to neural circuits in the brain of drosophila melanogaster, the common fruit fly. To discover these circuits, they require knowledge about the connectivity between neurons. Synapses are located at a neuron's arborisation. Where two or more neurons' arborisations overlap, they display a potential connectivity.

Neurobiologists use the information on potential connectivity to formulate new hypotheses and experiments. Visualising and computing overlaps between more than two arborisations (higher order overlaps) is difficult.



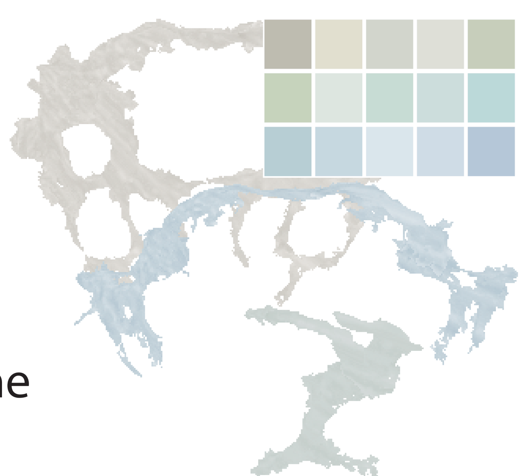
Two segmented neurons in BrainGazer

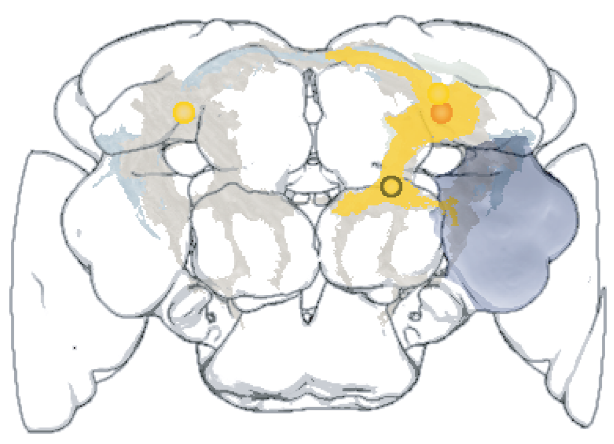
Data Visualization

With the goal to closely imitate the design, we chose to combine a variety of non-photorealistic rendering techniques.

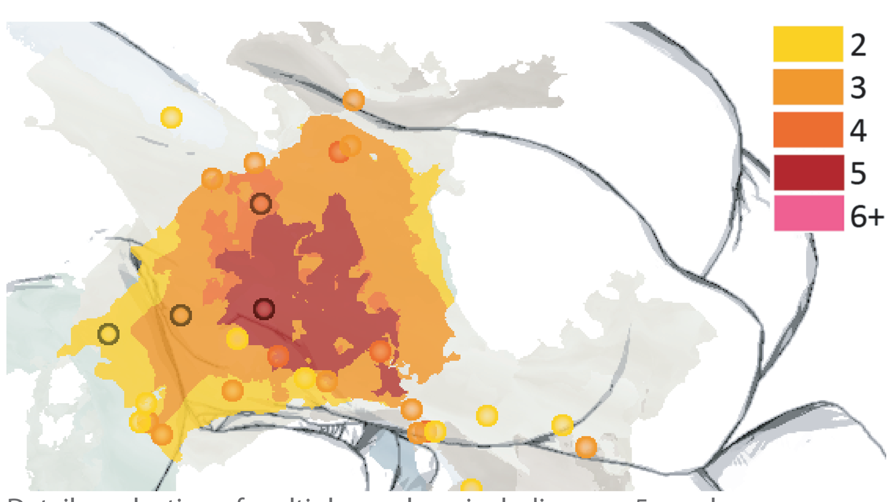

The silhouette of the brain

The 3D visualisation is a composite of five separate renderings. Among those are a silhouette rendering, a watercolour shader for arborisations, and a transparent rendering for optional neural structures. All these serve as context for the point of interest, the overlaps. These intersections of arborisations are rendered in bright colours (yellow pair-wise overlaps, orange shades for higher order overlaps). The visualisation shows overlaps on demand. They are abstracted as interactive glyphs. The glyphs (small circles) can be used to select/show overlaps or blend in the computed volumetric data. The shaders for overlaps and their glyph abstractions use A-Buffers on the GPU, the same data structure used for their volume computation (compare the box on the right).

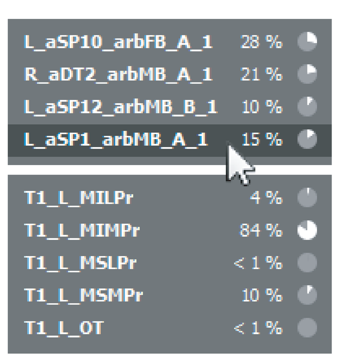

Arborisations and their colour palette


The final 3D visualisation shows a pair-wise overlap

The interactive 3D visualisation is complemented by two menus that display the computed volumetric data. Tooltips for glyphs are directly embedded into this 3D view. A separate menu in the viewport provides, in addition to the volumetric data, settings for the visibility of neuronal structures.

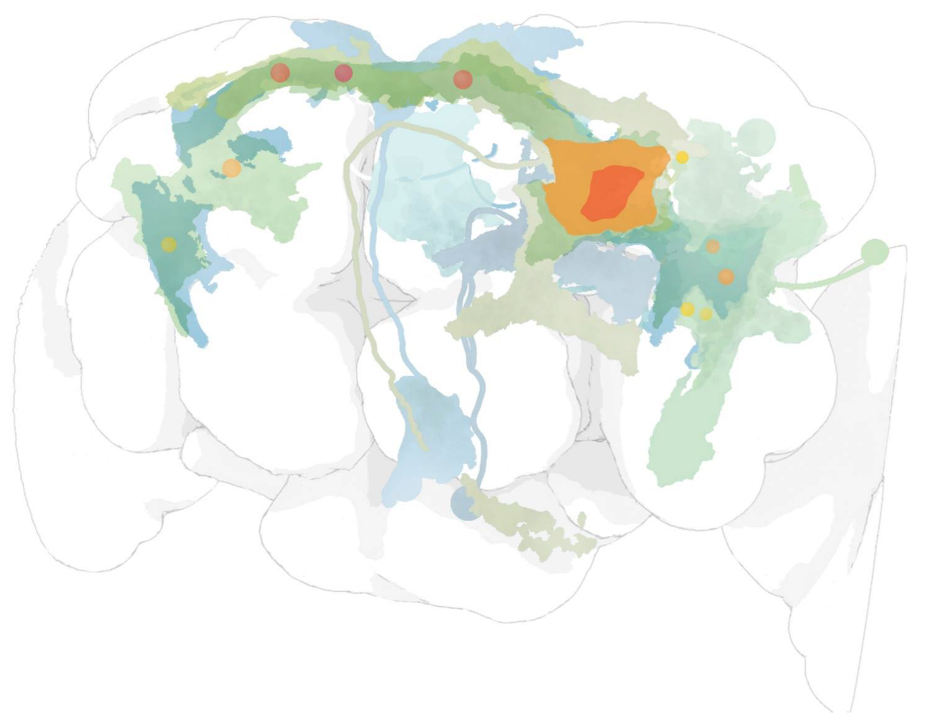


Detail: a selection of multiple overlaps, including one 5-overlap


Detail: a tooltip of a 4-overlap with enough volumetric data to judge the importance of the overlap

Motivation

We needed a visualisation for higher order overlaps (intersections of more than two arborisations). The artists Judith Moosburner at the Zürcher Hochschule der Künste developed a beautiful novel design in cooperation with the neurobiologists at the IMP in Vienna.

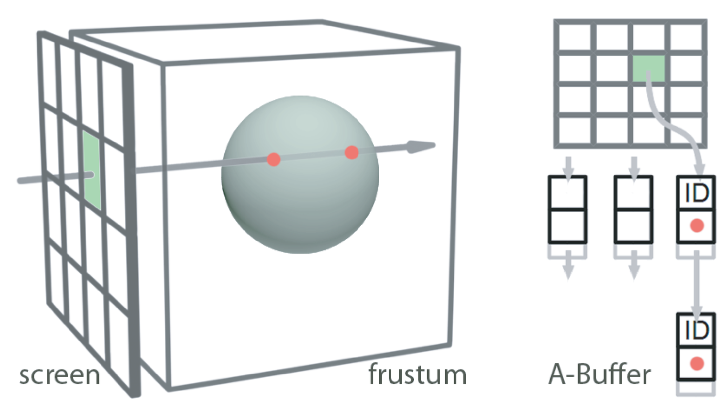

Visual design by Judith Moosburner

Goals of the Thesis

- Realise the novel design
- Compute volumes of higher order overlaps on demand
- Provide interaction techniques to communicate the data

Computing Intersections

A major contribution of the thesis is the algorithm to compute mesh volumes and mesh intersection volumes on the GPU. We use A-Buffers on the GPU—which are typically used to achieve order-independent transparency—to compute volumes. A framebuffer typically stores a single colour value per screen pixel. The A-Buffer, however, can record multiple values in a single screen pixel by allocating linked lists.



The graph below illustrates a simplified version of our algorithm, three passes on the GPU are suffcient. The first pass renders meshes and stores depth values (pictured as red dots on the right).

render meshes

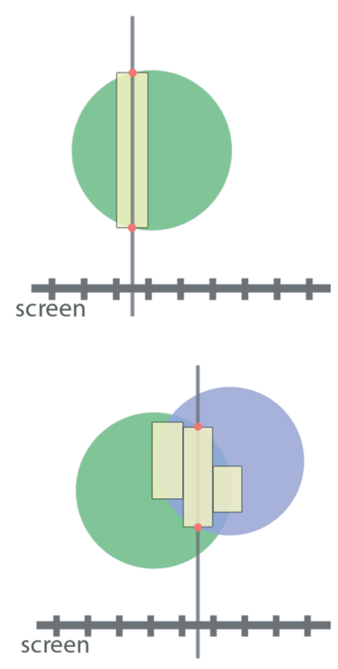
first A-Buffer

The rendering pipeline does not guarantee the order of fragments in each pixel. Thus, a compute shader sorts the depth values from the first A-Buffer and writes to a second A-Buffer.

sorting by depth

The second A-Buffer contains depth values sorted by depth. In the third and final pass, a compute shader iterates the linked list in each pixel to find depth values relevant for a specific volume.

second A-Buffer



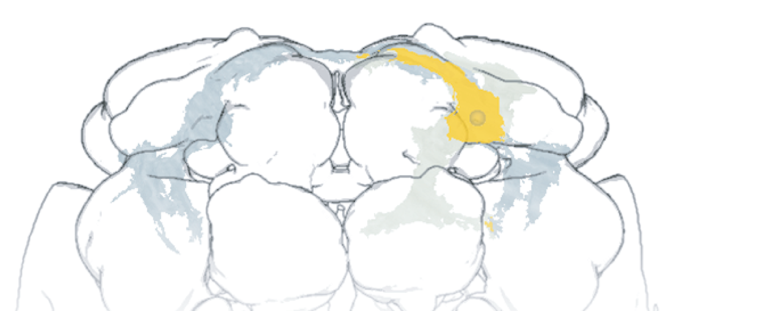
The examples on the left illustrate the depth differences in two dimension. In the simplest case we compute depth differences for a single mesh For a pair-wise overlap the algorithm finds the depth values of the intersection volume.

compute depth differences

The depth differences stored in the final A-Buffer can be summed up on the CPU to compute a volume.

final A-Buffer

Conclusion



- Features described by the design successfully implemented
- Very fast on demand computation of intersection volumes on the GPU
- Visual and quantitative investigation of higher order overlaps of arbitrary order
- Integrated with the BrainGazer framework used by neurobiologists
- Interactive links with other tools in BrainGazer

vrvis IMP Research Institute of Molecular Pathology