DOI: 10.1111/cgf.12792

COMPUTER GRAPHICS *forum*
Volume 36 (2017), number 1 pp. 160–171

Visualization and Quantification for Interactive Analysis of Neural Connectivity in *Drosophila*

N. Swoboda[1], J. Moosburner[2], S. Bruckner[3], J. Y. Yu[4], B. J. Dickson[5] and K. Bühler[1]

[1]VRVis Research Center, Vienna, Austria
nicolas.swoboda@gmail.com, buehler@vrvis.at
[2]Zurich University of the Arts, Switzerland
judith.moosburner@gmx.net
[3]University of Bergen, Norway
stefan.bruckner@uib.no
[4]Center for Integrative Neuroscience and Department of Physiology, University of California, San Francisco, CA, USA
jai@phy.ucsf.edu
[5]Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA
dicksonb@janelia.hhmi.org

Abstract

Neurobiologists investigate the brain of the common fruit fly Drosophila melanogaster to discover neural circuits and link them to complex behaviour. Formulating new hypotheses about connectivity requires potential connectivity information between individual neurons, indicated by overlaps of arborizations of two or more neurons. As the number of higher order overlaps (i.e. overlaps of three or more arborizations) increases exponentially with the number of neurons under investigation, visualization is impeded by clutter and quantification becomes a burden. Existing solutions are restricted to visual or quantitative analysis of pairwise overlaps, as they rely on precomputed overlap data. We present a novel tool that complements existing methods for potential connectivity exploration by providing for the first time the possibility to compute and visualize higher order arborization overlaps on the fly and to interactively explore this information in both its spatial anatomical context and on a quantitative level. Qualitative evaluation by neuroscientists and non-experts demonstrated the utility and usability of the tool.

Keywords: J.3 [Computer Applications]: Life and Medical Sciences—Biology and genetics

ACM CCS: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Discovering the relations between genes, neurons and behaviour is the key to gaining insights into how the brain works. *Drosophila melanogaster* is widely used as a model organism to discover and understand behaviour-specific circuits. The fly has a manageable number of neurons (around 100,000), exhibits complex behaviour and comes with a well-equipped toolbox for genetic dissection of anatomy and function of the nervous system [Gri12], [Sim09], [OW08]. Binary expression systems based on enhancer traps can be used to genetically label small groups of neurons (and even single neurons) and to monitor and modulate their activity. Their anatomy can then be visualized using confocal microscopy (see Figure 2, left). This technique has been used to identify circuits related to courtship behaviour [vPLY*11, YKD*10], the olfactory system [FW14], visual information processing [LCH*13] and walking direction [BMWD14], among others.

An individual invertebrate neuron consists of a **cell body** with one eventually branching **projection** ending in one or several **arborizations**. The brain is divided into 43 brain regions, the so-called **neuropils**, 'that synergistically cooperate to achieve computational tasks' [ISea14]. Arborizations are the branching structures at the end of a neuron's nerve fibres, where synapses communicate with other neurons.

The process of identifying which neurons form a circuit responsible for a certain behaviour is an iterative combination of experiments, anatomical screening and data annotation to explore spatial relationships between neurons that might comprise a specific circuit.

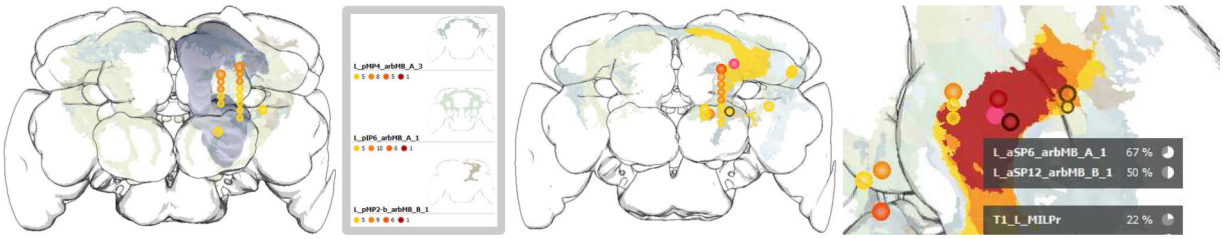


Figure 1: *Our tool visualizes overlaps of arborizations inside the brain of Drosophila melanogaster.*

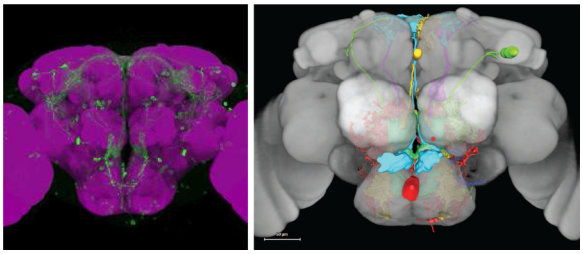


Figure 2: *Raw confocal microscopy image (left) and 3D rendering of standard brain with segmented neurons (right).*

The process culminates in further hypothesis building, subsequently triggering new experiments.

This analysis is based on spatial representations of neurons that have been annotated on double channel 3D confocal microscopy images showing binary expression patterns of neural cells and stained brain tissue [LCH*13], [LTW*11], [YKD*10]. As the images stem from different flies, all images are co-registered using the tissue channel to a standard brain using non-rigid registration. Thus, all segmented neuronal structures share a common spatial reference system and can be directly compared. This comparison is anatomically viable since many neurons in *Drosophila* are identifiable, i.e. the neural structures are highly stereotyped and can be identified in virtually all brains [OW08].

Due to limited spatial resolution and the fact that the neurons have been annotated on different brains, it is impossible to directly detect synaptic connectivity between neurons observed on two or more images. Instead, Peters' rule [PPW91] is used to formulate hypotheses about potential anatomical connectivity. This principle states that there is a direct relation between the existence of anatomical neuronal connectivity and overlapping arborizations. Overlap is a prerequisite for connectivity. This is supported by recent findings that neuronal connectivity can be estimated from axonal and dentric densities [vPvO13].

This study begins at the point where potential connectivity of a set of neurons must be analysed. Ideally, this should be possible in an interactive, efficient and effective manner, providing quick insight into higher order connectivity information and enough flexibility to react to new insights by quickly adding additional neurons.

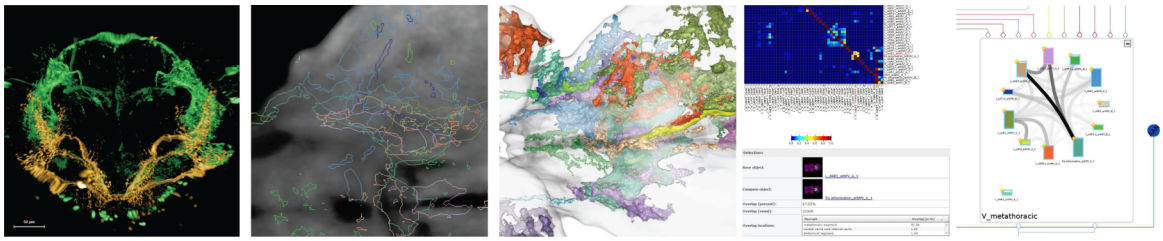
Direct 3D or sectional 2D display of neuron populations is currently only available in a very straightforward manner, without visual encoding of overlap or connectivity information, making intuitive detection and interactive exploration of the data difficult. If complex overlap patterns of several neurons are present, it becomes almost impossible to derive any detailed information from their direct spatial visual representations. Abstract representations containing quantitative information on potential connectivity, such as heat maps or graph-based representations like neuroMap [SBS*13], depend on precomputed pairwise arborization overlaps and do not support the analysis of higher order intersections for a flexible set of neurons (see Figure 3 and Subsection 1.1 for related tools).

The restriction of most interactive tools to pairwise overlaps is based on the fact that the number of possible combinations within large sets of neurons exhibit exponential growth. This makes precomputation of higher order overlaps of all neurons infeasible. Currently, this computation is done offline and for a static, preselected subset of neurons of interest. This lacks the flexibility of analysing changing sets of neurons in real time.

Our contribution The application described in this paper aims to fill a gap in the set of currently available tools for neural connectivity exploration. We present:

1. a *problem- and data-specific information and interaction design* worked out prior to implementation to create a design proposal unbiased by technical limitations.
2. an *implementation realizing the proposed design* by
   - introducing novel methods based on A-buffers to allow instant volumetric computation and glyph-based abstraction for arborization overlaps of arbitrary order,
   - applying state of the art GPU-based non-photorealistic rendering techniques,
   - linking the computed quantitative information via an interactive 3D visualization and two types of menus, providing different levels of abstraction,
3. a *qualitative evaluation* of our solution to determine its usability and usefulness, as well as improvements based upon the results followed by a second round of evaluations.

According to our collaborators, the proposed tool *allows computation and interactive exploration of higher order arborization overlaps for the first time*. It has great potential to accelerate hypothesis building in *Drosophila melanogaster* neural circuit research. This is an extended version of a paper originally presented at the VCBM Eurographics Workshop [SMB*14]. The extension adds more in-depth explanations as well as some further contributions:



(a) (b) (c) (d) (e)

Figure 3: *These images show different approaches to discovering potential anatomical connectivity. All images have been created using BrainGazer [BvG*09] and its plugins. (a) Overlay of two 3D staining expressions in a volume renderer showing overlapping arborizations. (b) A slice view showing (coloured) contours of segmented arborizations. (c) The same set of arborizations in a 3D rendering. Pairwise arborization overlaps can be quantitatively analysed and compared using (d) heat maps or (e) the graph representation neuroMap [SBS*13].*

1. a more robust algorithm for calculating glyph positions along with a new glyph layout,
2. multiple interactive user-guided filters to remove unimportant glyphs/overlaps and
3. a second round of evaluations specifically addressing the new changes.

1.1. Related work

Several data collections related to *Drosophila melanogaster* neural circuit research are publicly available. In relation to these collections, several tools for exploring images and annotated neurons are offered: Flylight [JRea12], Flybrain [SMO*11], Virtual Fly Brain [MOSR*12], FlyCircuit [CLea11] and BrainBase [Bra14]. Flylight and Flybrain provide confocal microscopic images, but no annotated neurons. The online portals of FlyCircuit and Virtual Fly Brain both provide graphically driven ontology queries over the set of available segmented neurons in the respective databases. They limit the (interactive) search for connectivity to finding neurons projecting to the same neuropils and do not explicitly detect or even quantify arborization overlaps. BrainBase provides precomputed overlap information for searches for overlaps in specific neuropils based on pairwise overlaps and parallel coordinates. All three portals provide 3D online rendering of detected neurons on demand, allowing visual inspection of the results. Offline tools like BrainGazer [BvG*09] and NeuronNavigator [LTW*11] provide high-quality 3D visualizations for neurons and sophisticated spatial query methods to find overlapping neurons at arbitrary positions. But none of the tools include any quantitative or qualitative information on the overlaps. neuroMap [SBS*13] provides an abstract view to explore potential connectivity with highly sophisticated encoding of precomputed overlap information of pairs of arborizations. Nevertheless, higher order overlaps are still difficult to detect and no quantitative information is available. Recently, Dercksen *et al.* [DEHO12] proposed a tool supporting structural connectivity analysis of a model of neuron populations in the barrel cortex. The tool helps to derive hypotheses about potential connectivity from the distribution of pre- and post-synaptic groups of neurons. This work comes closest to ours in intent, as it also combines 3D visualization with quantitative and qualitative elements, but the underlying data and therefore the methodology differ substantially from our solution.

2. Data, Workflows and Scientific Questions

Circuits that govern a particular behaviour in *Drosophila* are identified in a loop of behavioural experiments, genetic modulations and investigation of anatomical relationships of neurons. This leads to large collections of 3D double-channel confocal microscopy images of genetically modified flies generated using the GAL4/UAS system to highlight groups of neurons and nc82 staining to visualize brain tissue.

To make the anatomy of the imaged brains spatially relatable, all images are non-rigidly co-registered based on the nc82 staining. They are registered onto a standard brain generated from a carefully selected set of tissue images. The high stereotypy of this brain, as mentioned in Section 1, makes it possible to register all images to such a template image. All neuropils and the brain surface are annotated on the template image. Neurons are segmented and annotated according to the protocol described by Yu *et al.* [YKD*10]. To capture anatomical variations and improve on segmentation errors, they often segment images created from two or more scans of different flies of the same genetic line. The resulting cell bodies, arborizations and neuropils are stored as binary masks as well as individual geometric meshes; neuron projections are represented as centerline-radius information. All images and objects are stored in a relational database that is accessed by our system.

The overlap of arborizations is a necessary condition for the existence of anatomical connectivity, i.e. the presence of synapses. Volumes of arborizations are measured in μm^3 and so are overlaps. Scientists, however, judge the significance of an overlap not just on absolute volumes, but on volume ratios. The ratios between the volume of an intersection and the respective volumes of participating arborizations are key. According to Peters' rule (as mentioned in Section 1), the largest of these ratios, in percentage terms, is likely to be the most interesting to the user. A second point of interest is the distribution of the overlap among neuropils, i.e. which neuropils contain (at least parts of) the overlap and to what degree.

Detection and quantification of these overlaps is crucial in the process of formulating and testing hypotheses on behavioural circuits. With the data mentioned above available, neuroscientists are interested in three **core questions:**

1. **Which groups of neurons overlap?**
2. **Which neuropils contain (parts of) the overlap?**
3. **What is the significance of the overlap?**

Overlaps are usually analysed by looking at 2D slices and/or pre-calculated intersection volume data derived from 3D segmentation masks. The available tools restrict the scientists' workflows to visual inspection based on 2D sectional and 3D views and/or computation and pairwise comparison of arborization overlaps and their distribution to neuropils using simple spreadsheets, heat maps or recently introduced graph-based representations [SBS*13] (compare Figure 3 and Subsection 1.1 for relevant tools). To facilitate their daily analytical work, our collaborators specifically requested a way to overcome these limitations and interactively investigate higher order overlaps of several arborizations of different neurons inside a specific neuropil. And although the complexity grows exponentially with the number of overlapping arborizations, the solution must retain interactivity. This requirement pertains to both the retrieval of volumetric data and the 3D visualization. The principal idea governing our work is that the combination of good information and interaction design and online computation of overlaps can substantially improve the efficiency of this analytical process.

3. Information and Interaction Design

The information and interaction design was created in a closed feedback and discussion loop with a communication design professional and domain experts. The focus was on optimizing perceptual aspects of information flow to provide a design guideline for the realization of a tool to help neuroscientists answer the three questions posed in Section 2. Technical considerations were not central at this stage.

The brain and its neuropils provide the spatial reference system for all annotated neurons and related information. Together, these form an anatomical atlas of neuronal structures of the fly. The brain atlas resembles cartographic systems [Bre04] and the design work thus cleaved to concepts of cartography and information design, which particularly deal with ways to represent abstract and complex information [BGLL09], [KEBT10].

In the context of insights from perception and colour theory [Itt03], [KW05], interaction design [Spe07] and cartography [AH06], [Bre04], current 3D depictions of neurons (see Figure 3) exhibit substantial flaws: They suffer from excess detail, visual clutter, adverse colouring and missing connectivity representation.

In addressing the tasks outlined in Section 2, our design is grounded in the main principles of information design [BGLL09], [KEBT10]:

1. **Reduction:** to eliminate irrelevant information and focus on essentials, especially already existing visualizations and the representation of the brain.
2. **Abstraction:** to make connections visible, to identify clusters and core areas and to depict connectivity in order to highlight and rank it more quickly and easily.
3. **Information scaling:** to use interactivity to reduce the amount of information while still providing access to details on demand.

These topics were first approached by the designer through several artistic studies, leading from figurative to increasingly abstract representations (see Supporting Information), resulting in the final design discussed in the next paragraphs and shown in Figure 4.

3.1. Object, shape and colour design

Brain surface, neuropils and neurons provide a hierarchical context for the representation of connectivity information, providing support to answer the first and second core questions.

The available geometric representations of neurons, neuropils and brain surfaces were directly extracted from the 3D images, resulting in seemingly well-textured object surfaces. As the brain and its neuropils provide the spatial context for the neurons, these surfaces are widely reduced or even ignored. For the global context provided by the brain surface, a neutral white or grey with high transparency with enhancing silhouettes was chosen. Neuropils are depicted similarly, but with slightly more colour to establish differentiation.

Neurons provide vital contextual information for overlaps. The surfaces of the arborizations were particularly heterogeneous and structured, so they are reduced as much as possible in order for the details to not cause distraction. Slight texturing is used instead to obtain a lightweight effect of organic appearance of neurons. Projections appear as thin lines and cell body locations as spheres. For neuropils and neurons, a colouring scheme from the cold colour spectrum of brown/green/blue was chosen, because these colours are mostly observed as neutral. Neuropil colours were chosen from the unsaturated, pale-blue end of this spectrum.

Regions of overlap are defined by intersecting arborizations and strongly highlighted. A gradient colour scheme with intense red shades was specifically tailored to support the detection of higher order overlaps (core question 3). Pairwise overlaps are coloured in yellow, triple overlaps in orange and higher order overlaps in red and dark red. These bright, vivid colours have a high signalling effect and can be easily recognized and identified by their contrast to the rest of the system. Such gradient colour schemes and this signalling effect are popular instruments. Among others, the reasoning behind them is discussed in the literature on cartography [AH06], [Bre04].

3.2. Information scaling and interaction design

Using abstraction, information representation in interactive systems can be properly scaled [Spe07]. When depicting potential connectivity, information on the existence and significance of overlaps is the most important. Visualizing all overlap regions at once in 3D would be incredibly cluttered. To avoid this, the existence of an overlap is initially only displayed using a glyph in the form of a small dot indicating the order of overlap using the overlap colour scheme (Figure 4). These glyphs are easily recognizable within the contoured brain and are placed roughly at the position of the overlap. Clusters and core areas can thus be easily recognized.

The glyphs not only encode overlap information, they are also the central interaction element. Hovering over or selecting a glyph immediately visualizes its corresponding overlap using the overlap colour scheme. Clicking a glyph toggles its selection status on or off. Multiple overlaps can be selected this way at the same time (see

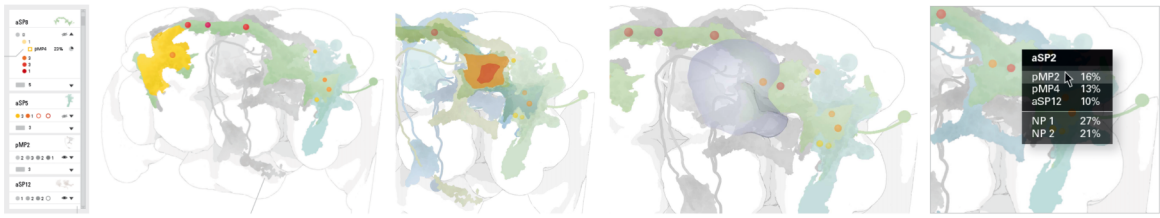


Figure 4: *Mock-ups representing the final design from left to right: (a) The tree menu bundles all quantitative information; its functionality is linked to other views. (b,c) Object, shape and colour design showing reduction and abstraction elements to visualize several context layers and enhancement of relevant information by applying appropriate colour schemes. (d) Compared to these flat abstractions, neuropils appear substantial. (e) A tooltip offers quantitative information about a single overlap with limited interaction.*

Figure 6 b for a selection of three overlaps). Individual overlaps can be deselected the same way. Clicking in the whitespace away from any glyphs undoes the entire selection immediately.

Two types of menu offer quantitative information on overlaps: the tree menu and a tooltip menu. This information consists of the list of participating arborizations and their relative volumes, i.e. how much of the arborization's volume is part of the overlap, as well as the distribution among neuropils. Investigating the quantitative data combined with interaction in 3D helps answer all three core questions.

The tooltip menu opens when mousing over a glyph (compare Figures 4 e and 6 c). It displays the aforementioned quantitative data and provides limited interaction. Moving the mouse over a line of information highlights the related arborization (or neuropil) in the 3D visualization. Only showing this additional spatial context on demand improves information scaling.

The second menu, the tree menu, located to the left of the 3D visualization (compare Figures 4 a and 6 a) compactly illustrates full details about all overlaps. This view is centred on the arborizations as roots of a tree structure. It is bi-directionally linked with the 3D visualization by employing the same colour and symbol encoding and mouse-over/select behaviour. The tree structure can be used to expand/collapse quantitative overlap information step by step. The tree menu allows the user to include neuropils of interest or highlight individual neurons, a toggle option lets them keep certain structures highlighted.

3.3. Expansion of the interaction design

Initial evaluations identified shortcomings in some areas of the design and its implementation [SMB*14]. This subsection discusses improvements to the design which address the more crucial test user requests: the glyph placement and the lack of filters.

The original design does not specify glyph positioning in detail. As a result of the frame-by-frame position calculations for glyphs, we encountered erratic glyph movements when, e.g. rotating the 3D visualization [SMB*14]. A new glyph layout groups glyphs vertically (compare Figure 6 b). The glyph groups stay in their relative positions when rotating the visualization; they can be optionally collapsed or expanded. The highest order glyph at this approximate position is always first; lower order glyphs at the same rough location are appended below.

The first round of evaluations also revealed that users wanted to filter overlaps they considered unimportant [SMB*14]. With this additional reduction of information, users expect to instantly get results for very specific research tasks. Thus, we extended the original interaction design to include user-guided filters for overlaps. To address all test user requests, we decided to design options to filter by order of overlap as well as by relative and absolute overlap volumes.

The new filters support information scaling by removing clutter. Unimportant overlaps and their glyphs can be interactively filtered out at any time; changes are applied immediately. It is the user's job to define—dependent on use case—what constitutes an unimportant overlap, although default settings filter out extremely small overlaps. An inconspicuous toolbox overlays the 3D visualization, with a legend relating the coloured dots to orders of overlap from two to six or more (compare Figure 6 b). Each glyph representation in the legend works as a check-box, where clicking toggles the filtering of all overlaps/glyphs of that specific order. Clicking on the toolbox symbol in the legend opens the volume filters, which use multiple double sliders to map volume ranges of overlaps (see Supporting Information). All filters may be combined. They are applied to the list of existing overlaps one after the other; as such they are linked with a logical AND.

4. Implementation

Our implementation realizes the information and interaction design proposed in Section 3 and provides support for the complete workflow related to the exploration of higher order arborization overlaps as described in Section 2. The implementation takes the need for interactive performance into account via highly efficient computational methods for volume calculation and interactive visualizations.

The tool has been integrated into a larger framework which includes classical 3D visualization, heat maps and graph-based representations for the exploration of pairwise overlaps. In addition to the original design, we also provide system-wide *cross-selection* of neuronal objects, allowing parallel investigation of the same data within different tools (see Supporting Information). The framework unites the different tools within a common workspace, listing all

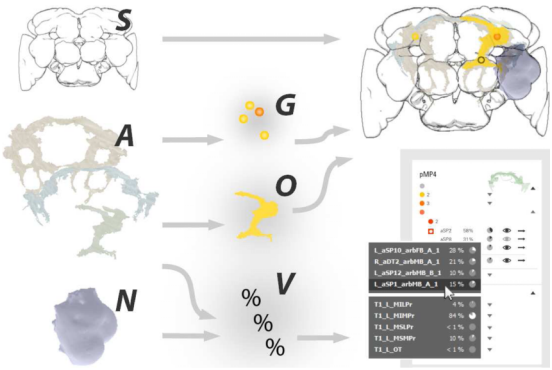


Figure 5: *From left to right, this image describes the pipeline. Mesh information on the left is used to create the output on the right. Arborization meshes, A, are needed to define overlaps, O, and glyphs, G. For the volume calculation, V, we need arborizations and neuropils, N. All these parts, along with the silhouette, S, contribute to the final rendering on the right. Quantitative information from the volume calculation can be explored in the menus, also depicted on the right.*

loaded arborizations. Across this workspace, selections are communicated to other tools. Thus, a selection in a heat map, which represents pairwise overlaps, will propagate to all other tools. In our tool, a selection is mirrored by selecting and therefore highlighting the appropriate pairwise overlap. This cross-selection has limitations, which we address in Section 7.2.

In the following, we describe in more detail the parts of the implementation which were not realized using standard techniques.

4.1. Computational pipeline

Figure 5 depicts a high level view of the pipeline. Once the neurons are loaded into the application, their arborizations undergo a volume estimation process. This calculates volumes of arborizations and neuropils and all arborization overlaps (Section 5). This information is stored and later used to feed the menus containing the quantitative information. Data loading and calculation take a few seconds, after which the rendering process starts. Loading additional arborizations initiates the volume estimation for the newly introduced overlaps.

The rendering process entails rendering context information like brain surface, neuropils and neurons, as well as drawing arborization overlaps and their glyphs (Section 6). For each feature, a specific shader renders into its own texture; eventually, these are blitted together in an alpha compositing step. Keeping these textures from one frame to the next saves valuable render time. If, e.g. an additional neuropil has to be rendered, only one shader has to update one texture. Other more complex shaders such as rendering overlaps, which requires an A-buffer to be resolved, are skipped.

4.2. Basic data structure

Although the neuronal structures exist as segmentation masks, we decided to calculate intersection volumes from their mesh representation, because available memory limits the number of binary segmentation masks we can load at a time.

Choosing the right data structure is crucial to achieving the required interactive performance. The structure has to deliver instant computation of arborization intersections and provide interactive performance for visualization of loaded and computed data. Good candidates for data structures handling larger numbers of overlapping meshes are the G-buffer [KMS07] and the A-buffer [Car84]. Nevertheless, as we must not filter out any depth information, the G-buffer would require a separate texture for each mesh, far too many for our requirements. To create exact representations of intersection meshes, a CSG tree could be used. This would result in exact volumes; however, even with efficient modern implementations, the process of building a CSG tree and the subsequent boolean operations takes up to a few minutes for a single intersection.

Based on these considerations, we decided to employ A-buffers as a basic data structure. We use a set of A-buffers to calculate the volumes of arborizations and intersections (see Section 5) and a separate set of A-buffers for the interactive visualization of overlap information (see Subsection 6.1). A-buffers can be efficiently implemented on the GPU using OpenGL 4.x. Our implementation is derived from the one by Crassin [Cra10] using C++ with glsl shaders.

5. Quantification of Arborization and Overlap Volumes

This step robustly calculates the absolute volumes for all arborizations of loaded neurons, the absolute volumes of all arborization intersections, absolute neuropil volumes and absolute volumes of arborization intersections with neuropils. The calculations are performed using A-buffers on the GPU. In the final step, relative overlap volumes (with respect to a specific arborization) and the distribution of overlap regions to neuropils are calculated on the CPU.

At least two major shader steps are necessary to perform the volume estimation. First, meshes are rendered and stored to the A-buffer. Second, per-pixel mesh depths are written to the CPU, where their sum, the estimated volume, is calculated.

5.1. Storing mesh data in the A-buffer

Meshes are rendered to the A-buffer using an orthogonal projection matrix fitted to the scene, in this case the brain's bounding box. In the first shader step, they are projected against the z-axis, while the original z-values are stored in the A-buffer along with mesh identifiers. We do not cull back faces as we need all depth information later.

5.2. Calculating depth differences from the A-buffer

At this stage of our pipeline, all meshes have been rendered to the A-buffer and are therefore available in global GPU memory. Each pixel indexes its own linked list of mesh information. A simple convex mesh would create two entries in each pixel it occupies: one data point for the front face and one for the back face. A data point records the depth (the linear view-space depth, in this case, a z-value

in model space) of the mesh location in 3D-space and a mesh ID to identify which arborization the depth entry belongs to.

Before working on the A-buffer, its stored values are sorted by depth. In a pixel, a single mesh has a depth of $d = d_{\text{out}} - d_{\text{in}}$, where d_{in} is the first depth value in the A-buffer and d_{out} is the second. More complex, non-convex meshes may have a multiple of two depth entries in a single pixel. After sorting, these $2n$ depth values are alternating entry (d_{in}) and exit (d_{out}) points to the mesh interior. Thus, the depth d_{pix} of a mesh in a single pixel of the A-buffer is:

$$d_{\text{pix}} = \sum d_{\text{out}} - \sum d_{\text{in}} = \sum_{i=0}^{n-1} d_{2i+1} - \sum_{i=0}^{n-1} d_{2i}. \quad (1)$$

For each mesh, these depth differences are written to a float buffer and later totalled by the CPU. The sum of all depth differences is a representation of the mesh volume. Using $width_{bb}$, $height_{bb}$, the width and height of the brain's bounding box, used to create the projection matrix, and $width_{ab}$, $height_{ab}$, the A-buffer's width and height, the estimated mesh volume is

$$V_{\text{estimated}} = \frac{width_{bb} \cdot height_{bb}}{width_{ab} \cdot height_{ab}} \sum d_{\text{pix}}. \quad (2)$$

5.3. Intersecting meshes in the A-buffer

The fact that the A-buffer offers data pairs of depth and mesh identifier, sorted by depth value, allows boolean operations to be performed on the meshes. In our case, we only need intersections of meshes. Note that this description applies to the intersection $A \cap B$ of two meshes A and B. Extending it to higher order overlaps or intersections between $A \cap B$ and neuropil meshes is trivial.

The per-pixel iteration of the A-buffer can be interpreted as a simple ray casting. While the ray moves through the linked list of depth values, the shader continuously updates a list to track mesh entries and exits. This way it can decide between mesh interior and exterior at any point along the ray.

If, at a particular point, both meshes A and B have been entered, the depth value must be an entry point for their intersection. Depth values for exit points are determined analogously. The shader then calculates depth differences and stores them to a buffer to be totalled by the CPU later.

5.4. Overlap ratio calculations

Calculating the sums of the depth difference buffers from the previous steps could be achieved by reduction on the GPU. Since it did not turn out to be a bottleneck (compare t_4 in Table 1), we do this on the CPU to arrive at the absolute volumes.

From there, still on the CPU, we derive percentages describing the relation between a specific arborization and a given overlap region. A value of 100% means the arborization lies entirely inside the investigated overlap; a value of 0% means the arborization and the overlap are disjoint.

Table 1: *The brain's bounding box used to fit the orthogonal projection matrix has width 420 and height 315. These are the volume estimation results of 50 arborization meshes, with over 8 million triangles in total. The first column shows the resolution of the A-buffer. $\bar{r}$ and s_r, respectively, denote the mean and variance of the estimation's divergence from an exact calculation. The timings (in ms) are approximate upper limits on an NVIDIA GeForce GTX 670 and an Intel Core i7 920. t_1: allocate A-buffer memory and render meshes, t_2: calculate depth differences on GPU, t_3: download buffer of depth differences to CPU, t_4: sum on CPU (the t_1 step may have to be repeated if insufficient memory is initially allocated on the GPU).*

ine Size	$\bar{r}$	s_r	t_1	t_2	t_3	t_4
ine ine 64x64	0.9887	0.0777	16	5	2	1
ine 128x128	1.0039	0.0281	16	17	4	1
ine 256x256	1.0030	0.0080	21	58	14	6
ine 512x512	0.9999	0.0033	26	73	44	24
ine						

The distributions of overlap volumes to neuropils are also calculated. Necessary depth differences from intersections between $A \cap B$ and all neuropils are also calculated on the GPU (see Subsection 5.3).

5.5. Accuracy

Calculating the per-pixel depths, d_{pix}, and adding them up to $V_{\text{estimated}}$ introduces a negligible numerical error. In an actual use case, the number of loaded arborizations is likely limited to about 10. For accuracy testing purposes, we use $i = 50$ arborization meshes, rendered to an A-buffer with a 512 by 512 resolution. The resolution of the A-buffer controls the number of depth differences that a particular mesh is divided into. Generally, more samples in the x- and y-directions make for a more exact estimation.

We calculate the triangle mesh volumes $V_{\text{calculated}}$ as a ground truth using signed volumes of polyhedrons. Dividing each $V_{\text{calculated}}$ by the corresponding $V_{\text{estimated}}$ results in 50 ratios, r_i, which we expect to each be 1. The test leads to a mean, $\bar{r} \approx 0.999934$, and a variance, $s_r \approx 0.003300$. This method of estimating volumes proves more than sufficient for our use case. Even an A-buffer size of 128 by 128 turns out to be sufficiently accurate in practice. Compare Table 1, which lists timings and accuracy of different resolutions.

6. Visualization Methods

The following subsections describe some of the rendering and visualization techniques used to approximate the design while maintaining interactive performance. They are parts of the computational pipeline (Subsection 4.1, Figure 5). Overlaps are rendered using an A-buffer. The menus to explore quantitative data use Qt4.

6.1. Connectivity information

The essential information we need to encode in the 3D rendering is the location and order of overlaps. The abstraction choice of using glyphs makes it easy to visualize many overlaps at a time.

Loading multiple neurons with intersecting arborizations creates an exponential amount of overlaps, each abstracted by its own glyph.

Rendering meshes to the A-buffer. As in the volume estimation, we employ an A-buffer to store all mesh data. The visualization should support 3D interaction, so the rendering step uses an appropriate perspective projection.

The volume estimation renders to an A-buffer with an orthogonal projection to achieve useful depth samples for an exact calculation. The A-buffers used here, however, store depth values distorted by the perspective projection. To find all intersections for rendering, it is sufficient that the depth order is maintained.

Rendering overlaps from the A-buffer. To render an overlap, our shader needs only the A-buffer containing the mesh data and the IDs of the overlapping meshes. This is simpler than the volume estimation on intersections; here, the shader only needs to iterate the A-buffer to discover if the overlap in question exists in a pixel at all. Except for a subtle transparency, the overlaps receive no special shading. The colouring is done according to the design; the colour indicates the order of the overlap. Note that overlaps of higher order are usually rendered on top, since they occupy less area (compare Figure 6 b). The performance is comparable to common uses of A-buffers (e.g. for order-independent transparency), as our shader essentially performs a single iteration of the A-buffer.

Rendering glyphs. To remain individually discernible, glyphs ought to appear on top of the overlap they represent while keeping their distance to other glyphs. The first attempt, positioning in screen space, led to issues with temporal coherence. Instead, we opted for positioning in 3D space and a new layout.

A glyph is placed where its overlap produces the highest response in the depth differences buffer used by the volume estimation. This results in multiple (related) glyphs at the same position. When rendering in screen space, one glyph is rendered there while all others at the same position are appended vertically (compare Subsection 3.3 and Figure 6 b). Optionally, this layout collapses to show only the first glyph; on mouse-over the rest are revealed.

6.2. Rendering of the context information

The silhouette of the brain is calculated by applying a Sobel filter to a depth buffer [ST90] after a simplified mesh of the brain template has been rendered to it. The filter response is used to create a silhouette ranging from dark to light grey, with non-continuous transitions between three grey values. The method achieves real-time performance and close resemblance to the initial design. The neuropils are rendered as transparent bluish surfaces (Figure 1, left).

The colouring-style and texture chosen for the neurons resemble watercolour images. Looking for an interactive technique that avoids the shower door effect, we found the technique implemented by Bousseau *et al.* [BKTS06] very useful. We use their colour modification to apply two of the watercolour effects they describe: low-frequency turbulent flow and high-frequency pigment dispersion. This results in a rendering with limited depth and structure information. As intended, the arborizations look flat and unobtrusive, keeping the visual focus on the overlaps.

One arborization can be highlighted in the 3D visualization at a time, rendering it darker to make it stand out. Its cell body is also rendered, abstracted as a dot. The dot is placed in the centre of the cell body bounding box. Also, the arborization's projection is drawn with constant diameter. Figures showcasing these abstract visualizations in both the design and the implementation may be found in the Supporting Information.

6.3. User interfaces for quantitative analysis

Glyphs are the central interaction element; interactions are responsive with virtually no delay. As required by the design (compare Subsection 3.2), glyphs provide mouse-over and select actions and they are linked to the two menus (Figures 6 a and c). All interaction interfaces beside the glyphs were implemented with Qt4.

The glyphs open the tooltip menu on mouse-over. It displays the quantitative data related to a specific glyph/overlap (compare Figure 6 c for a quadruple overlap). The tooltip menu's mouse-over actions—highlighting a single arborization or rendering a single neuropil—are very responsive, also due to the selective updates to the necessary textures (compare Subsection 4.1) before the alpha compositing, e.g. only updating the rendered neuropils.

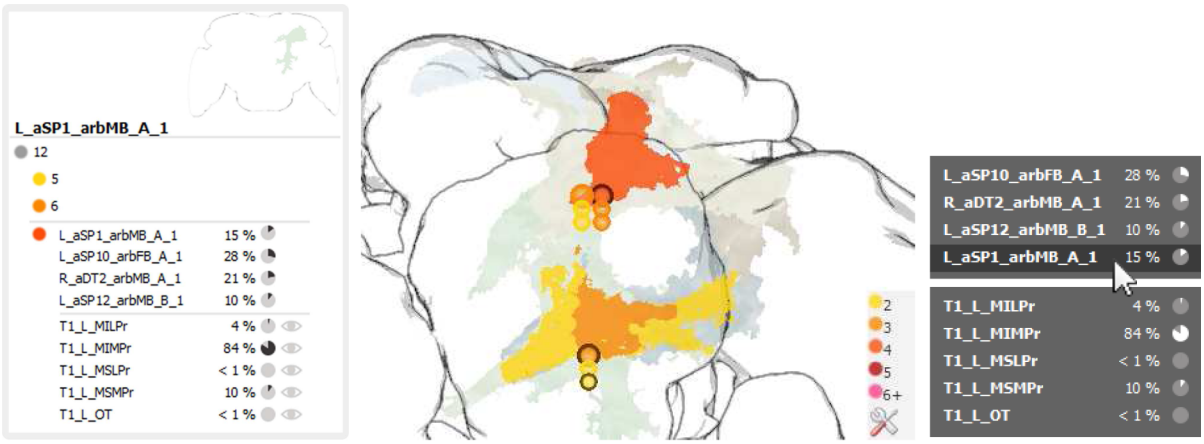
The second menu, the tree menu, located to the left of the 3D visualization (compare Figure 4 a left and Figure 6 a), compactly illustrates complete details on all overlaps. The menu has the same mouse-over/select behaviour as the rendering. The bi-directional link with the 3D visualization is again accelerated by selectively updating textures in the rendering process. The displayed quantitative data are the same in both menus. The volume estimation provides enough accuracy to show these percentages without a decimal place.

The quantitative analysis is supported by a filter interface (compare the design in Subsection 3.3). Changes in the filter are immediately propagated to the rendering. This mostly involves filtering out some glyphs and updating their texture with the glyph billboard shader. Again, this produces virtually no delay.

7. Qualitative Evaluation

We performed a qualitative evaluation of the software to gain insight into its usability and usefulness by assessing effectivity, efficiency and user satisfaction.

Test subjects. The test group consisted of five experts and five non-expert users. Two of the five experts were highly experienced post-docs with a strong background in neural circuit research. Three expert users had a bioimage informatics and visualization background and decent knowledge of workflows related to neural circuit research. Non-experts had no background in neuroscience, but varying experience of user interfaces and 3D tools in general. The test users were not involved in the development, except for one expert user, who contributed in the beginning of the design phase.



(a) (b) (c)

Figure 6: *The tree menu (a) is expanded to show details of a quadruple overlap (the relative volumes for the participating arborizations, below the overlap volume's distribution among the neuropils). The tooltip menu (b) provides the same data directly inside the 3D visualization via mouse-over. The 3D visualization (b) shows six pastel-coloured arborizations, producing nine overlaps. Three glyphs have been selected, revealing their respective overlaps. The triple overlap is rendered on top of the pairwise overlap since the latter fully encloses the triple overlap. Overlaps have a very flat look, as intended by the design; their volumetric nature is made apparent when rotating the view. Also, when rotating, the first glyph in a group—here each with three glyphs—stays in the same relative position, while their appendices stay vertically aligned below. The legend in the right lower corner (b) provides the filter functionality for overlap order; it expands to show volume filter options (compare the Supporting Information).*

Test setup. Tests were performed in front of a computer in our lab running the system. All test subjects received a brief introduction into the test setup and—if necessary—the application background for the tools. Users could freely interact with the system and were guided by a set of tasks and questions and—if necessary—hints on how to proceed.

7.1. Evaluation process

The users were confronted with a simple pairwise overlap and told to try out the user interface on their own. Then, they moved on to a workspace with multiple arborizations, creating quadruple overlaps. Both these test scenes were previously defined to provide unified starting points for all users. During and after the evaluation, they were asked detailed questions about the user interface and general questions about clarity and efficiency. For non-expert users, we did not explain the domain extensively, but simply stated that they were to look at overlaps between meshes inside the brain. The users were not tasked with specific goals, instead we followed the 'think-aloud' method [LR93] to capture users' thoughts and feelings while freely interacting with the entire system. This method includes occasional hints to guide the user (e.g. 'What do you think these percentages mean?' or 'What behaviour did you expect when you clicked there?'). On agreement from the test subject, the interview was recorded for later transcription.

Two expert and five non-expert users re-tested the system in a second round of evaluation to produce feedback on recent changes, in particular glyph layout and filters. The following paragraphs, unless stated otherwise, refer to results from the first round of evaluations.

7.2. User feedback

Connectivity exploration. All expert users stated that our tool accelerates both finding and analysing new overlaps compared to workflows they were previously familiar with. In general, the experts were enthusiastic about the connectivity exploration for higher order overlaps. Non-experts were able to understand spatial relations between overlaps and arborizations and most called the exploration intuitive, except for a few issues pointed out in the following paragraphs.

Visual design and colour scheme. The colour scheme and overall visual design were highly praised. According to most users, the focus and context visualization achieved by the reduced representations of brain, neuropils and neurons perfectly supports the search for higher order overlaps. One expert user criticized the colouring of arborizations as too faint but especially liked the arborization highlighting. The highlighting with cell body and projection was appreciated by the expert users, but, of course, required explanation for non-experts. Most agreed that the colour scheme for glyphs is intuitive. All but one non-expert user understood the colour coding after investigating glyphs for a short time.

Glyph interaction. All users considered interacting with glyphs intuitive. From the first round of evaluations, we learned that users were irritated by the erratic movement of glyphs when rotating the view. This issue stemmed from the glyph position calculation in screen space [SMB*14] and has since been solved by calculating positions in 3D view space.

As part of the second round of evaluations, users tested the vertical glyph layout (compare Subsection 3.3 and Figure 6 b, for

implementation details see Subsection 6.1). Users considered the grouped layout tidier, and liked that the 3D placement virtually removes occlusions between glyphs. All users welcomed the temporal coherence when rotating the 3D visualization. Some assumed a hierarchical relationship within groups, which is not guaranteed based on the implementation but was observed in practice. One non-expert wanted to keep the old layout as an alternative.

Menu interaction Both menus were considered to be generally clear in the first round of evaluations. One non-expert did not find them intuitive at all and recommended opening the tooltip menu like a context menu by right clicking. Some users from both groups wanted the process of expanding and collapsing the tree menu to be controlled by arrows. One expert user and, as expected, most non-experts were confused by the percentages listed in the menus.

Selection model. The users found their way around the selection and deselection of multiple overlaps (compare Subsection 3.2) via menu or 3D visualization very quickly, despite some having different expectations. One non-expert assumed multiple selection would work by pressing the control button while clicking and one expert did not like that selecting more than one overlap at once was even possible. These two users criticized that cross-selection with other tools in the framework, e.g. overlap heat maps, was only done with one overlap at a time, as this is inconsistent with multiple selection inside the tool. The users liked that the tree menu and 3D visualization are linked and consistent in using the same dot symbols for overlaps.

Overlap filters. During the first round of evaluations, users requested filtering techniques to remove clutter by what they considered unimportant overlaps (compare Subsection 3.3 and the Supporting Information). In the second round of evaluations, users gave feedback on the utility of the filters. Filtering by overlap order was immediately understood and appreciated; this may be attributed to the simple check-box approach. No further explanations were needed for this filter. Two non-experts expected mouse-over behaviour for these check-box buttons to preview changes. The more complex volume-range filters allow the user to interactively define which overlaps are unimportant. These range sliders for relative or exact volume ranges required explanation for non-experts, who were again confused by relative volumes. As these filters were a specific request in the first round of evaluations, it comes as no surprise that the users welcomed these improvements. Expert users considered the filters sufficient to ask and answer specific research questions about overlap ratios and order.

7.3. Discussion

It is the opinion of all users familiar with comparable workflows that our tool speeds up and improves the analysis of higher order overlaps. All expert users were able to answer the three core questions (posed in Section 2) on number, location and significance of overlaps by investigating using the 3D visualization and the integrated menus. As expected, non-experts only found answers to core questions one and two. This positive feedback is especially valuable from two expert users from the field of neurocircuitry, who see a clear benefit in using the proposed tool compared to their daily workflows.

The selection model lacks clarity, especially with respect to system-wide cross-selection of neurons or overlaps over different tools, a feature not defined in the design (compare Section 4). Although warmly welcomed by most users, the cross-selection is limited by other tools to the selection of only a single pairwise overlap or a single neuron. In the future, integrating these limitations with the selection of multiple higher order overlaps in our tool may be better achieved by more clearly emphasizing selections in the tree menu. However, the fact that other tools do not support selections of multiple pairwise overlaps at the same time—let alone selections of higher order overlaps—will remain a limiting factor for cross-selection.

The vertical glyph layout replaced the old layout calculated in screen space [SMB*14]. Although this is a major improvement, we are investigating further changes. This, combined with filtering the layout, eliminates cluttering as a result of too many glyphs. In some borderline instances, this is not enough. The default method of collapsing the glyph groups—showing only the first, bigger glyph—has the drawback of hiding vital information. Covering all use cases while keeping the layout simple and intuitive is a subject for future work.

The completely different perspective on topic and data contributed by the designer at the beginning of the project enriched the technical and biological views of the rest of the team in a sustainable manner. Not limited by technical concerns, the graphic designer was free to imagine a novel visualization, bringing together their domain knowledge with the biologists' requirements. The positive feedback on the design encourages us to continue this strategy of separate design and implementation processes in the future.

8. Conclusion

In this paper, we presented a new design and its implementation to enable analysis of overlaps of arbitrary order. The qualitative evaluation supports our statement that this tool is both effective and efficient. The design and implementation were developed in collaboration with domain experts. In the course of the evaluation, these neuroscientists confirmed that the visualization and interaction features achieve the tool's aim of supporting hypothesis formulation. It does this by answering neuroscientists' questions on number, location and significance of overlaps. For the first time, scientists have a tool to calculate this potential connectivity between multiple neurons and interact with it in an uncluttered spatial context.

We expect neuroscientists to not only use the tool for analysis and exploration of overlaps, but also to communicate findings. Future efforts will explore options to extend the glyph interaction used as abstraction for overlaps. This may include implementation of improved filters that employ advanced interactive visualization techniques or extended encodings: e.g. overlap importance as glyph size.

Acknowledgement

The authors thank the test subjects and László Tirián for valuable contributions. This work was funded by the FFG Headquarters Project 'Molecular Basis' Grant number: 834223.

References

[AH06] Abrams J., Hall P.: *Else/Where: Mapping New Cartographies of Networks and Territories*. University of Minnesota Design Institute Minneapolis, Minneapolis, Minnesota, USA, 2006.

[BGLL09] Bohnacker H., Gross B., Laub J., Lazzeroni C.: *Generative Gestaltung*. Verlag Hermann Schmidt, Mainz, Germany, 2009.

[BKTS06] Bousseau A., Kaplan M., Thollot J., Sillion F. X.: Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of NPAR* (2006), pp. 141–149.

[BMWD14] Bidaye S. S., Machacek C., Wu Y., Dickson B. J.: Neuronal control of Drosophila walking direction. *Science 344*, 6179 (2014), 97–101.

[Bra14]Brainbase. Institute of Molecular Pathology Vienna, 2014. http://brainbase.imp.ac.at. Accessed 1 November, 2014.

[Bre04] Brewer C.: *Designing Better Maps: A Guide for GIS Users*. Environmental Systems Research, Redlands, California, USA, 2004.

[BvG*09] Bruckner S., Šoltészová V., Gröller M. E., Hladůvka J., Bühler K., Yu J., Dickson B.: Braingazer - Visual queries for neurobiology research. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1497–1504.

[Car84] Carpenter L.: The A-buffer, an antialiased hidden surface method. In *Proceedings of SIGGRAPH* (1984), pp. 103–108.

[CLea11] Chiang A.-S., Lin C.-Y., Chuang, C.-C., Chang, H.-M., Hsieh, C.-H., Yeh, C.-W., Shih, C.-T., Wu, J.-J., Wang, G.-T., Chen, Y.-C., Wu, C.-C., Chen, G.-Y., Ching, Y.-T., Lee, P.-C., Lin, C.-Y., Lin, H.-H., Wu, C.-C., Hsu, H.-W., Huang, Y.-A., Chen, J.-Y., Chiang, H.-J., Lu, C.-F., Ni, R.-F., Yeh, C.-Y., Hwang, J.-K.: Three-dimensional reconstruction of brain-wide wiring networks in Drosophila at single-cell resolution. *Current Biology 21*, 1 (2011), 1–11.

[Cra10] Crassin C.: Opengl 4.0+ abuffer v2.0: Linked lists of fragment pages. http://blog.icare3d.org/2010/07/opengl-40-abuffer-v20-linked-lists-of.html, 2010. Accessed August 13, 2013.

[DEHO12] Dercksen V. J., Egger R., Hege H.-C., Oberlaender M.: Synaptic connectivity in anatomically realistic neural networks: Modeling and visual analysis. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (2012), pp. 17–24.

[FW14] Fişek M., Wilson R. I.: Stereotyped connectivity and computations in higher-order olfactory neurons. *Nature Neuroscience 17*, 2 (2014), 280–288.

[Gri12] Griffith L. C.: Identifying behavioral circuits in Drosophila melanogaster: Moving targets in a flying insect. *Current Opinion in Neurobiology 22*, 4 (2012), 609–614.

[ISea14] Ito K., Shinomiya K., Ito M., Armstrong J. D., Boyan G., Hartenstein V., Harzsch S., Heisenberg M., Homberg U., Jenett A., Keshishian H., Restifo L., Rössler W., Julie H. S., Nicholas J. S., Roland S., Leslie B. V.: A systematic nomenclature for the insect brain. *Neuron 81*, 4 (2014), 755–765.

[Itt03] Itten J.: *Kunst der Farbe. Studienausgabe. Subjektives Erleben und objektives Erkennen als Wege zur Kunst*. Urania, Stuttgart, 2003.

[JRea12] Jenett A., Rubin G. M., Ngo T.-T., Shepherd D., Murphy C., Dionne H., Pfeiffer B. D., Cavallaro A., Hall D., Jeter J., Iyer N., Fetter D., Hausenfluck J. H., Peng H., Trautman E. T., Svirskas R. R., Myers E. W., Iwinski Z. R., Aso Y., DePasquale G. M., Enos A., Hulamm P., Lam S. C., Li H. H., Laverty T. R., Long F., Qu L., Murphy S. D., Rokicki K., Safford T., Shaw K., Simpson J. H., Sowell A., Tae S., Yu Y, Zugates C. T.: A GAL4-driver line resource for Drosophila neurobiology. *Cell Reports 2*, 4 (2012), 991–1001.

[KEBT10] Klanten R., Ehmann S., Bourquin N., Tissot T.: *Data Flow 2 - Visualizing Information in Graphic Design*. Gestalten Verlag, Berlin, Germany, 2010.

[KMS07] Kalkofen D., Mendez E., Schmalstieg D.: Interactive focus and context visualization for augmented reality. In *Proceedings of ISMAR* (2007), pp. 1–10.

[KW05] Klant M., Walch J.: *Grundkurs Kunst, Band 1-3*. Schroedel Verlag, Braunschweig, Germany, 2002, 2003, 2005.

[LCH*13] Lin C.-Y., Chuang C.-C., Hua T.-E., Chen C.-C., Dickson B. J., Greenspan R. J., Chiang A.-S.: A comprehensive wiring diagram of the protocerebral bridge for visual information processing in the Drosophila brain. *Cell Reports 3*, 5 (2013), 1739–1753.

[LR93] Lewis C., Rieman J.: *Task-Centered User Interface Design: A Practical Introduction*. A shareware book published by the authors, 1993.

[LTW*11] Lin C.-Y., Tsai K.-L., Wang S.-C., Hsieh C.-H., Chang H.-M., Chiang A.-S.: The neuron navigator: Exploring the information pathway through the neural maze. In *Proceedings of IEEE PacificVis* (2011), 35–42.

[MOSR*12] Milyaev N., Osumi-Sutherland D., Reeve S., Burton N., Baldock R. A., Armstrong J. D.: The virtual fly brain browser and query interface. *Bioinformatics 28*, 3 (2012), 411–415.

[OW08] Olsen S. R., Wilson R. I.: Cracking neural circuits in a tiny brain: New approaches for understanding the neural circuitry of Drosophila. *Trends in Neurosciences 31*, 10 (2008), 512–520.

[PPW91] Peters A., Palay S. L., Webster H. D.: *The Fine Structure of the Nervous System: Neurons and Their Supporting Cells*. Oxford University Press, Oxford, UK, 1991.

[SBS*13] Sorger J., Bühler K., Schulze F., Liu T., Dickson B.: neuromap - interactive graph-visualization of the fruit fly's neural circuit. In *Proceedings of IEEE BioVis* (2013).

[Sim09] Simpson J. H.: Mapping and manipulating neural circuits in the fly brain. *Advances in Genetics 65* (2009), 79–143.

[SMB*14] Swoboda N., Moosburner J., Bruckner S., Yu J. Y., Dickson B. J., Bühler K.: Visual and quantitative analysis of higher order arborization overlaps for neural circuit research. In *Proceedings of VCBM 2014* (2014), pp. 107–116.

[SMO*11] Shinomiya K., Matsuda K., Oishi T., Otsuna H., Ito K.: Flybrain neuron database: A comprehensive database system of the Drosophila brain neurons. *Journal of Comparative Neurology 519*, 5 (2011), 807–833.

[Spe07] Spence R.: *Information Visualization: Design for Interaction*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2007.

[ST90] Saito T., Takahashi T.: Comprehensible rendering of 3-D shapes. In *Proceedings of SIGGRAPH* (1990), pp. 197–206.

[vPLY*11] von Philipsborn A. C., Liu T., Yu J. Y., Masser C., Bidaye S. S., Dickson B. J.: Neuronal control of Drosophila courtship song. *Neuron 69* (2011), 509–522.

[vPvO13] van Pelt J., van Ooyen A.: Estimating neuronal connectivity from axonal and dendritic density fields. *Frontiers in Computational Neuroscience 7* (2013), 160.

[YKD*10] Yu J. Y., Kanai M. I., Demir E., Jefferis G. S. X. E., Dickson B. J.: Cellular organization of the neural circuit that drives Drosophila courtship behavior. *Current Biology 20*, 18 (2010), 1602–1614.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher's web site:

Figure S1: Design studies.

Figure S2: Object, Shape and Color Design.

Figure S3: Connectivity and Interaction Design.

Figure S4: Implementation Details: Tree Menu and Tooltip.

Figure S5: Implementation Details: two arborizations with a pairwise overlap on the left; three arborizations with a triple overlap and three pairwise overlaps on the right.

Figure S6: Implementation Details: To the left: a pairwise overlap partially occluded by a triple overlap, a quadruple overlap above.

Figure S7: Implementation Details: a single neuropil in semi–transparent blue provides additional spatial context to the arborizations and overlaps.

Figure S8: Implementation Details: Above: A screen–shot of our tool in the BrainGazer software shows it side–by–side with other tools, here the graph representation neuroMap and an overlap heat–map.

Video S1.