Eurographics Workshop on Visual Computing for Biology and Medicine (2014)
I. Viola, K. Bühler, and T. Ropinski (Editors)

# Visual and Quantitative Analysis of Higher Order Arborization Overlaps for Neural Circuit Research

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
[2]Zurich University of the Arts, Switzerland
[3]University of Bergen, Norway
[4]Center for Integrative Neuroscience and Department of Physiology, University of California, San Francisco, United States
[5]Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, Virginia, United States

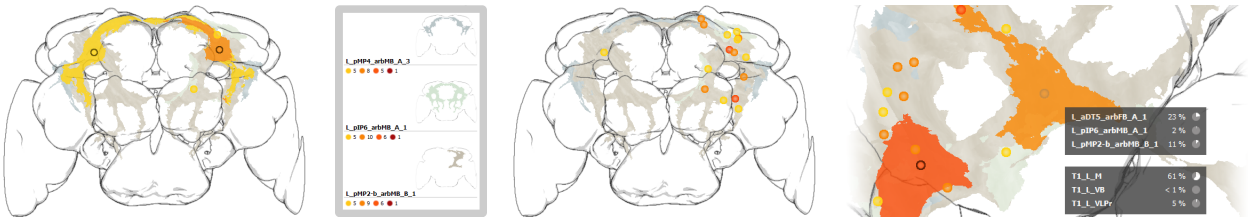



**Figure 1:** *Our tool visualizes overlaps of arborizations inside the brain of Drosophila melanogaster.*

**Abstract**

*Neuroscientists investigate neural circuits in the brain of the common fruit fly Drosophila melanogaster to discover how complex behavior is generated. Hypothesis building on potential connections between individual neurons is an essential step in the discovery of circuits that govern a specific behavior. Overlaps of arborizations of two or more neurons indicate a potential anatomical connection, i.e. the presence of joint synapses responsible for signal transmission between neurons. Obviously, the number of higher order overlaps (i.e. overlaps of three and more arborizations) increases exponentially with the number of neurons under investigation making it almost impossible to precompute quantitative information for all possible combinations. Thus, existing solutions are restricted to pairwise comparison of overlaps as they are relying on precomputed overlap quantification. Analyzing overlaps by visual inspection of more than two arborizations in 2D sections or in 3D is impeded by visual clutter or occlusion. This work contributes 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 its spatial anatomical context and on a quantitative level. Qualitative evaluation with neuroscientists and non-expert users demonstrated the utility and usability of the tool.*

Categories and Subject Descriptors (according to ACM CCS): J.3 [Computer Applications]: Life and Medical Sciences—Biology and genetics; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—

## 1. Introduction

Discovering the relations of genes, neurons and behavior is the key to gain insights in how the brain works. *Drosophila melanogaster* is used by a broad community as model organism to discover and understand behavior specific circuits. The fly has a still manageable number of neurons (approx. 100.000), exhibits complex behavior and comes with a well equipped toolbox for genetic dissection of anatomy and function of the nervous system [Gri12,Sim09,OW08]. Using enhancer trap-based binary expression systems it is possible to genetically access small groups of neurons, to visualize their anatomy using confocal microscopy and to monitor

and modulate activity (see figure 2, left). This technique is used to identify circuits related for example to courtship behavior [vPLY*11, YKD*10], the olfactory system [FW14], visual information processing [LCH*13] and walking direction [BMWD14].

The process of identifying neurons forming a circuit responsible for a certain behavior is a loop of experiments, anatomical screening and data annotation. This incorporates exploration of spatial relationships of neurons that might comprise a specific circuit. The process culminates in further hypothesis building, triggering subsequently new experiments.

Basis for this analysis are 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.

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 for hypothesis building on potential anatomical connectivity. This principle states that the existence of anatomical neuronal connectivity and overlapping arborizations are in direct relation to each other. Overlap is a prerequisite for connectivity. It is supported by recent findings that neuronal connectivity can be estimated by axonal and dentric density fields [vPvO13].

The work presented in this paper starts at the point when potential connectivity of a set of neurons has to be analyzed. Ideally this should be possible in an interactive, efficient and effective manner, providing fast insight into higher order connectivity information and enough flexibility to react to new insights by quickly adding additional neurons.

Nevertheless, direct 3D or sectional 2D display of neuron populations is currently only available in a very straight forward manner without visual encoding of overlap or connectivity information making intuitive detection and interactive exploration of the data difficult. Especially if complex overlap patterns of several neurons are present, it is almost impossible to derive any detailed information from their direct spatial visual representations. Abstract representations containing quantitative information on potential connectivity, like 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).

The restriction of most interactive tools to pairwise overlaps is based on the fact that large sets of neurons exhibit

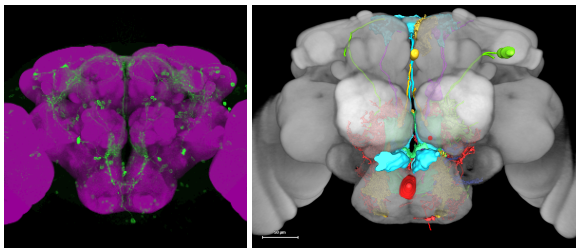


**Figure 2:** *Raw confocal microscopy image on the left and 3D rendering of standard brain with segmented neurons to the right.*

an exponential growth of possible combinations. 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. It lacks the flexibility to provide analysis of changing sets of neurons in real time.

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

- a *problem and data tailored information and interaction design* worked out prior to implementation to receive a design proposal unbiased by technical limitations.
- an *implementation realizing the proposed design* by
  - introducing novel A-Buffer based methods allowing 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 quantitative computed information between an interactive 3D visualization and two types of menus, providing different levels of abstraction
- a *qualitative evaluation* of our solution to determine its usability and usefulness.

According to our collaborators the proposed tool *allows for the first time to compute and interactively explore higher order arborization overlaps on the the fly* providing great potential to accelerate hypothesis building in neural circuit research of *Drosophila melanogaster*.

### 1.1. Related work

Several data collections related to neural circuit research of *Drosophila melanogaster* are publicly available. In relation with 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, Flybrain and Flybase provide only confocal microscopic images, but no annotated neurons. The online portals of FlyCircuit and

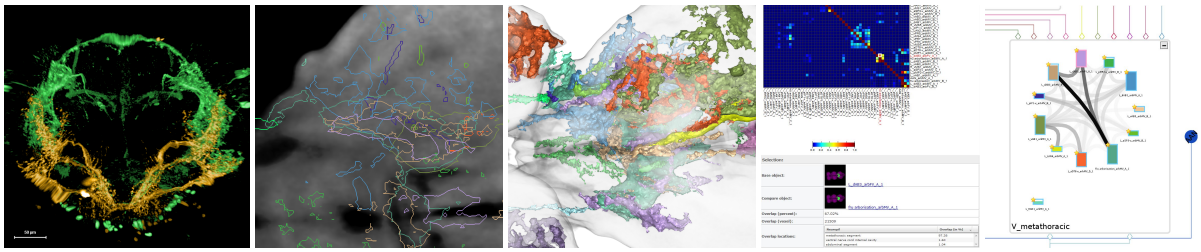



**Figure 3:** *These images show the most common tools to discover potential anatomical connectivity. From left to right: Overlay of two 3D staining expressions showing overlapping arborizations, slice view showing contours of segmented arborizations, 3D rendering of the same set of arborizations, heat map of pairwise arborizations overlaps, graph representation of overlaps*

Virtual Fly Brain both provide graphically driven ontology queries over the set of available segmented neurons in the respective data bases. 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 pairwise overlaps and parallel coordinates based search for overlaps in specific neuropils. All three portals provide 3D online rendering of detected neurons on demand, allowing a 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, also here higher order overlaps are difficult to detect and no quantitative information is available. Recently a tool supporting structural connectivity analysis of a model of neuron populations in the barrel cortex has been proposed by Dercksen et al. [DEHO12]. Here a hypothesis on potential connectivity is derived from the distribution of pre- and postsynaptic groups of neurons. From its intention, this work comes closest to ours, as it combines also 3D visualization with quantitative and qualitative elements, but the underlying data and therefore also the methodology differs substantially from our solution.

## 2. Data, Workflows and Scientific Questions

A neuron of an invertebrate consists of a **cell body** with one eventually branching **projection** ending in one or several **arborizations**. The brain is divided in 43 so called **neuropils**, "that synergistically cooperate to achieve computational tasks" [ISea14].

Circuits that govern a certain behavior in the drosophila are identified in a close loop of behavioral 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 enable spatial relation of the imaged brains all images are non-rigidly co-registered based on the nc82 staining to a standard brain, generated from a carefully selected set of tissue images. Neurons of interest are segmented and annotated. All neuropils and the brain surface were annotated on the template image. After segmentation, cell body, projection and arborizations of neurons, neuropils and the standard brain surface are available as binary masks and separate geometric mesh or, in case of projections, 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 $\mu m^3$ and so are overlaps. The scientists, however, judge the significance of an overlap not solely on absolute volumes, but on volume ratios. The essential numeric overlap values are the ratios between the volume of the intersection and the respective volumes of participating arborizations. According to Peters' Rule (as mentioned in the introduction), the greatest of these ratios, displayed in percentages, is likely to be the most interesting to the user. A second value of interest is the distribution of the overlap among neuropils.

Detection and quantification of these overlaps is a major step for hypothesis building on behavioral circuits. Having the above mentioned data available, neuroscientists are interested in getting a fast answer to 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?**

The current practice in analyzing arborization overlaps is restricted 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 excel sheets, heat maps or recently introduced graph based representations [SBS*13]. Overcoming this limitation

and getting the possibility to interactively investigate also higher order overlaps of several arborizations of different neurons inside a specific neuropil was an explicit request by our collaborators to facilitate their daily analytical work. We face the challenge of answering the three questions for higher order overlaps. Although the complexity grows exponentially with the number of arborization, the solution must provide interactive performance. The idea constituting the guideline for our work is that the combination of a good information and interaction design and online computation of overlaps can substantially alleviate this analytical process.

## 3. Information and Interaction Design

The information and interaction design was created in a close feedback and discussion loop between a communication design professional and domain experts. Focus was on optimizing perceptional aspects of information flow with the goal to provide a design guideline for the realization of a tool supporting neuroscientists in answering 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. All items together form an anatomical atlas of neuronal structures of the fly. The brain atlas resembles cartographic systems [Bre04] and the design work thus heavily aligned to concepts of cartography and information design, which deals especially with ways of representing abstract and complex information [BGLL09, KEBT10].

In the context of insights from perception and color theory [Itt03, KW05], interaction design [Spe07] and cartography [AH06, Bre04], current 3D depictions of neurons (see figure 3) reveal substantial flaws: They suffer from too much detail, visual clutter, adverse coloring, and missing representation of connectivity.

Addressing the tasks outlined in section 2, our design is grounded on the main principles of information design [BGLL09, KEBT10]:

- **Reduction** to decimate dispensable information and to confine information to its essentials, especially focusing on already existing visualizations and the representation of the brain.
- **Abstraction** to make connections visible, to identify clusters and core areas and to depict connectivity in order to highlight and rank it faster and more easily.
- **Information scaling** through interactivity to reduce the amount of information while still providing access to details on demand.

In a first step, these topics have been approached by the designer through several artistic studies, leading from figurative to more and more abstract representations (see supplemental material), resulting in the final design discussed in the next paragraphs and shown in figure 4.

**Object, Shape and Color Design** Brain surface, neuropils and neurons are providing a hierarchical context for the representation of connectivity information, providing support to answer the first and second core question.

The available geometric representations of neuron, neuropil and brain surface have been 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 gray with high transparency with enhancing silhouettes, was chosen. Neuropils are depicted similarly, but with slightly more color to establish differentiation.

Neurons provide the contextual information for overlaps - considered to be the central information for the users. 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. Instead slight texturing is used 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 coloring scheme from the cold color spectrum of brown / green / blue, colors which are mostly observed as neutral, was chosen. Neuropil colors were chosen from the unsaturated, pale end of the spectrum.

Regions of overlap are defined by intersecting arborizations and highlighted - in opposition to the contextual information - in an eye catching manner. As the detection of higher order overlaps is in the center of interest, a gradient coloring system with intense colors was designed supporting the detection of significant overlaps (core question 3). It colors pairwise overlaps in yellow, triple ones in orange, and higher order overlaps in red and dark red. The bright, vivid colors have a high signaling effect and can be easily recognized and identified by their contrast to the rest of the system. Gradient coloring schemes are also used in cartography to depict growing values, e.g. temperature.

**Information Scaling and Interaction Design** Using abstraction, information representation in interactive systems can be properly scaled [Spe07]. Most important in depicting potential connectivity is the information on the existence and significance of overlaps. Visualizing all overlap regions at once in 3D would lead to a completely cluttered view. To avoid this, the existence of an overlap is initially only displayed using a glyph in form of a small dot indicating the order of overlap using the overlap color scheme (figure 4). The glyphs are easily recognizable within the contoured brain and are placed at a rough 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 a glyph immediately visualizes its corresponding overlap using the

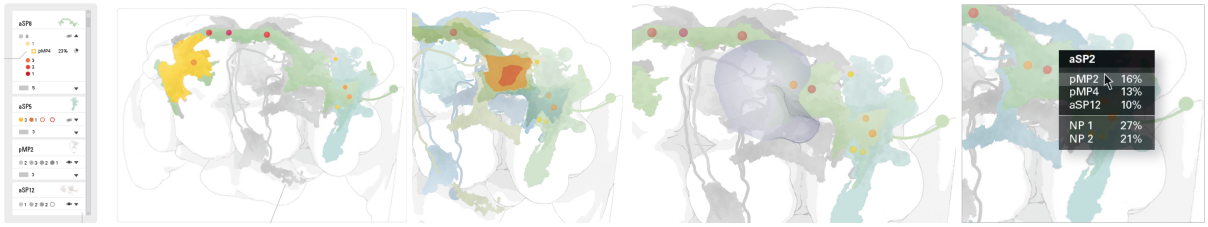



**Figure 4:** *The final design from left to right: (1) The tree menu bundles all quantitative information, its functionality is linked to other views. (2,3) Object, shape and color design showing reduction and abstraction elements to visualize several context layers and enhancement of relevant information by application of appropriate color schemes. (4) Compared to these flat abstractions, neuropils have a look of substance. (5) A tooltip offers quantitative information for a single overlap with limited interaction.*

overlap color scheme. Clicking the glyph makes this a permanent selection, revealing the overlap in the same way. Multiple overlaps may be selected at a time. Two types of menus offer quantitative information on overlaps. The described hover action opens a tooltip menu, displaying information on neurons and neuropils involved in the overlap and related quantitative data (compare figure 6(b) for a four-overlap). The second menu, the tree menu, located to the left of the 3D visualization (compare figure 4, left) compactly illustrates complete details on all overlaps. This view is centered on the arborizations as roots to a tree structure. It uses the same coloring scheme and symbols as the 3D visualization, to provide easy transferability and linking between the views. It offers the same hover/select behavior as the glyphs. The tree structure can be used to expand/collapse quantitative overlap information step by step. Both the tree menu and tooltips allow blending in neuropils of interest or highlighting single neurons. Investigating the quantitative data combined with interaction in 3D helps find answers to all core questions.

## 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 tool has been integrated into a larger framework including classical 3D visualization, heatmaps 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. The implementation takes the need for interactive performance into account. This requirement is mirrored in the choice of highly efficient computational methods for volume calculation and interactive visualizations.

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

### 4.1. Computational Pipeline

Figure 5 depicts a high level view on the pipeline. Once the neurons are loaded into the application, their arborizations undergo a volume estimation process. It 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. This entails rendering of context information like brain surface, neuropils and neurons, as well as drawing arborization overlaps and rendering their representing glyphs (section 6). Loading additional arborizations initiates the volume estimation for the newly introduced overlaps.

### 4.2. Basic Data Structure

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

The choice of the right data structure to handle this data is critical to achieve the required interactive performance for our application. The data structure to be chosen has to be able to deliver instant computations of arborization intersections as well as 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, by 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 boolean operations take up to a few minutes for a single intersection.

Based on these considerations we decided to use a set of A-Buffers as a basic data structures. As described in the next

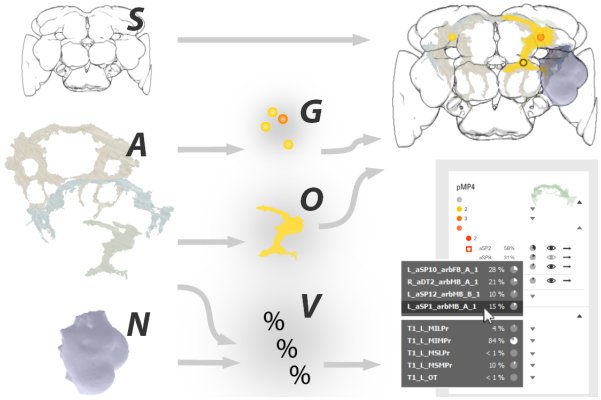



**Figure 5:** *From right to left, this image describes the pipeline. Mesh information on the left is used to create the output on the right. Arborization meshes A are needed to define overlaps O and glyphs G. For the volume calculation V, we need arborizations and neuropils N. All these parts, along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

two sections the A Buffer is allowing us to realize both, arborization and overlap volume calculation and interactive visualization of overlap information. A-Buffers can be efficiently implemented on the GPU using OpenGL 4.x. Our implementation is derived from that of 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. In a last step relative overlap volumes (in 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.

**Storing mesh data in the A-Buffer.** Meshes are rendered to the A-Buffer using an orthogonal projection matrix, which is 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 on.

**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 our case of the orthogonal projection this is a z-value in model space) of where the mesh is located in 3D-space and, of course, 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 the depth of $d = d_{out} - d_{in}$, 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

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

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

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

**Intersecting meshes in the A-Buffer.** The fact that the A-Buffer offers data pairs of depth and mesh identifier, sorted by depth value, permits performing boolean operations on the meshes. In our use case, we only need intersections of meshes.

Calculating depth differences (and subsequently volumes) of an intersection requires knowledge about which depth value represents an entry or exit to the intersection interior. For a pairwise overlap, i.e. an intersection of two meshes $A$ and $B$, the shader needs only their two respective mesh identifiers as they are stored in the A-Buffer alongside the depth values.

While iterating the A-Buffer, the shader continuously updates a list of meshes that have been entered. If, at a point, both meshes $A$ and $B$ have been entered, the depth value must be an entry point for the intersection. This way entry and exit points are determined and eventually, as described above for simple meshes, the depth differences are written to a buffer to be later summed up by the CPU.

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

**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, of all together over 8 million triangles. The first column shows the resolution of the A-Buffer. $\bar{r}$ and $s_r$ denote mean and variance of the estimation's divergence from an exact calculation. The timings (in* ms*) are approximate upper limits on a 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 at first not enough memory was allocated on the GPU)*

This shader step described here may be preceded by creating an additional A-Buffer storing mesh IDs of overlaps. This optional step of creating and downloading an Overlap-Buffer is described in section 6.1. It serves to decide on the CPU, which mesh intersections actually exist. This way, the shader calculating the depth differences is only ever sent mesh ID combination creating sensible intersections.

**Overlap Ratio Calculations** From the absolute volumes calculated in the previous steps we derive percentages describing the relation of a specific arborization to a given overlap region. The value of 100% means, that the arborization lies entirely inside the investigated overlap, a value of 0% means that the arborization and the overlap are disjoint. Furthermore the distributions of overlap volumes to neuropils is calculated.

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

As ground truth we calculate the triangle mesh volumes $V_{calculated}$ using signed volumes of polyhedrons. Dividing each $V_{calculated}$ by the corresponding $V_{estimated}$ results in 50 ratios $r_i$, which we expect to be 1 each. The test leads to a mean $\bar{r} \approx 0.999934$ and a variance $s_r \approx 0.003300$. The method of estimating volumes proves by far sufficient for our use case. Compare table 1, listing timings and accuracy of different resolutions.

## 6. Rendering techniques

The following subsections describe the rendering techniques used to achieve an approximation to the design while maintaining interactive performance. They are eventually combined to a final rendering as depicted in figure 5. The silhouette and neuropil meshes directly contribute to this final image. Rendering of arborizations, overlaps and glyphs requires the use of an A-Buffer.

### 6.1. Connectivity Information

The essential information we need to encode in the 3D rendering is the location and size 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.

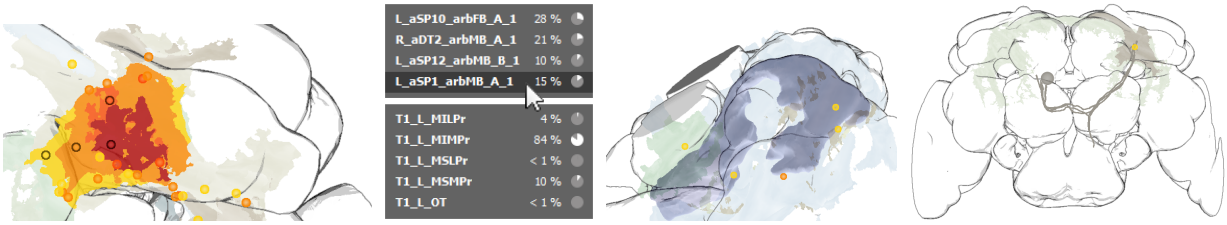
**Rendering meshes to the A-Buffer** As in the volume estimation, we employ an A-Buffer to store all mesh data. The visualization should support 3D interaction, so the rendering step uses an appropriate perspective projection.

**Creating and downloading an Overlap-Buffer** In a separate render stage, after the A-Buffer has been filled with all arborization meshes, we render its contents to a second A-Buffer, which we call Overlap-Buffer. In this step, we combine meshes to overlaps. The resulting Overlap-Buffer stores in each pixel a list of overlaps that exist there. In the volume estimation, this step is optional to determine which overlaps exist. We use it there, to limit the calculation of depth differences to only those intersections that actually bound a positive volume. Here, we download this Overlap-Buffer to the CPU to later use it to calculate glyph positions.

**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 meshes constituting the overlap. More simply than the volume estimation on intersections, this shader needs only 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 coloring is done according to the design, the color 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(a)).

**Rendering Glyphs** Concerning the glyphs, it was necessary to find a suitable way to position them in screen space. They ought to appear on top of the overlap they represent while keeping their distance to other glyphs, to stay discernible from each other. Additionally, there should be some sort of coherence when navigating the rendering by zoom or rotation.

We find glyph positions by calculating a center point in 2D for each overlap. First, the above mentioned Overlap-Buffer containing the two-dimensional overlap data is downloaded from the GPU. The small buffer is quickly iterated on




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

**Figure 6:** *(a) shows four rendered overlaps (of order 5, 4, 3, and 2). Notice that the glyph of the four-overlap is placed only on the part that extends the smaller five-overlap. (b) shows the tooltip of a four-overlap. (c) shows a neuropil with slight transparency. In (d), one of two neurons is highlighted, its cell body, abstracted as a gray dot, and its projection are rendered.*

the CPU and, by applying a city block distance transform, each overlap's center point is determined.

For each overlap, we create a separate distance field. Here, an overlap is considered *inside* only at pixels where it is not fully enclosed by an overlap of immediate higher order. This achieves aesthetically placed glyphs (figure 6(a)). After these distance calculations on the CPU, the glyph positions are rendered by a billboard shader, with slight transparency. Colors refer to the order of the respective overlap.

While rotating the view of the brain to explore the overlaps in more detail, glyphs will jump erratically from one location to another. This highly distracting behavior results from calculating centroids in 2D screen space instead of 3D view/projection space. To alleviate this problem, we interpolate glyph positions between frames. When changing the camera view, glyphs lag behind minimally. Within 600ms of the last movement, they reach their calculated positions. Over this time, the movement is negatively accelerated.

### 6.2. Rendering of the Context Information

The silhouette of the brain is calculated by applying a Sobel filter on a depth buffer, as suggested by Saito and Takahashi [ST90]. A simplified mesh of the brain template, on which all neuronal data has been registered, is rendered to a texture. Its depth data is filtered to decide where silhouettes occur. The filter response is used to create a silhouette ranging from dark to light gray, with non-continuous transitions between three gray values. The method achieves real-time performance and close resemblance to the initial design. The neuropils are rendered as transparent bluish surfaces (figure 6(c)).

The coloring-style and texture chosen for the neurons resemble watercolor 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 color modification to apply two of the watercolor effects they describe, these are 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 to keep the visual focus on the overlaps.

A single arborization at a time can be highlighted in the 3D visualization, which renders it a little bit darker to make it stand out. Additionally its cell body is rendered, abstracted as a dot (see figure 6(d)). The dot is placed in the center of the cell body bounding box. Also, the arborization's projection is drawn with constant diameter.

## 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 Persons** The group of test persons consisted of five experts and five non-expert users. Two of the five experts were highly experienced postdocs with strong background in neural circuit research. Three expert users had a bioimage informatics and visualization background and decent knowledge on workflows related to neural circuit research. Non-experts had no background in neuroscience, but varying experience of user interfaces and 3D tools in general.

**Test Setup** Tests were performed in front of a computer in our lab running the system. All test persons 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 where given the task to investigate a simple pairwise overlap and try out the user interface on their own. Then, they moved on to a workspace with multiple arborizations, creating four-overlaps. Both these test scenes were previously defined to provide unified starting points for all users. During the evaluation and afterwards, they where asked detailed questions on the user interface and overall

questions about perspicuity and efficiency. For non-expert users, we did not explain the domain extensively, but simply stated they were to look at overlaps between meshes inside the brain. We followed the "think-aloud" method [LR93] to capture thoughts and feeling of the test persons while interacting with the system. On agreement by the test person the interview was recorded for later transcript.

### 7.2. User Feedback

**Connectivity Exploration** As expert users stated, analysis of overlaps is commonly done by looking at 2D slices and/or precalculated intersection volume data derived from 3D segmentation masks. (Both of these work-flows can not reasonably be applied to higher order overlaps.) Their unanimous impression is that our tool accelerates both finding new overlaps and analyzing them.

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

**Interaction** Interacting with glyphs was considered intuitive by all users. Three expert users were irritated by erratic movement of glyphs when rotating the view. Of those, one addressed glyphs occluding each other. Very small overlaps are not immediately discernible in the 3D visualization when selecting them in the tree menu, as one non-expert user mentioned. Three expert users want this solved by filtering out unimportant overlaps altogether, e.g. by manually selecting a percentage threshold. Two more experts would like to filter by order of overlap. Both menus were considered overall perspicuous. One non-expert did not find them intuitive at all and recommends opening the tooltip menu like a context menu by right clicking the mouse. Some users from both groups would like the process of expanding and collapsing in the tree menu to be guided 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 via menu or 3D visualization very quickly, despite different expectations of some. 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 heatmaps, was only done with one overlap at a time as this is inconsistent with multiple selection inside the tool. The users liked that tree menu and 3D visualization are linked and consistent in using the same dot symbols for overlaps.

### 7.3. Discussion

It is the opinion of all users, that our tool speeds up and improves the analysis of higher overlaps, compared to different workflows. In this regard, all expert users were able to answer the three core questions (posed in section 2) on number, location and significance of overlaps by investigation both the 3D visualization and the integrated menus. The positive feedback is specially valuable coming from two users who are experts in the domain of neurocircuitry using different workflows in their daily life and seeing a clear benefit in using the proposed tool.

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. Although very 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. Integrating these limitations with the selection of multiple higher order overlaps in our tool may be better achieved in the future by more clearly emphasizing selections in the tree menu. Another field of improvement revealed by the evaluation concerns the glyphs jumping in between frames. Temporal coherence with glyph positioning will have to be addressed more thoroughly. Connected to this is the issue of glyphs rendered partially on top of each other. This sometimes occurs with overlaps of order four or above, although zooming into such a cluster alleviates the problem. A possible solution in addition to filtering unimportant glyphs (as requested by some users) may be dislocation via a simple 2D particle system.

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. The positive feedback on the design encourages us continue with this strategy of separated design and implementation process 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. These neuroscientists confirmed, in the course of the evaluation, that the visualization and interaction features achieve the tool's aim: to support hypothesis formulation. It does this by answering the neuroscientists' questions on number, location and significance of overlaps. For the first time, the scientists have a tool to calculate this potential connectivity

between multiple neurons, and interact with it in an uncluttered spatial context.

We expect the 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, that is used as abstraction for overlaps. This includes smart filtering techniques, better temporal coherence for glyph positioning, and encoding e.g. overlap importance as glyph size.

**Acknowledgment** This work has been funded by the FFG Headquarter 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, 2006. 4

[BGLL09] BOHNACKER H., GROSS B., LAUB J., LAZZERONI C.: Generative gestaltung. *Verlag Hermann Schmidt* (2009). 4

[BKTS06] BOUSSEAU A., KAPLAN M., THOLLOT J., SILLION F. X.: Interactive watercolor rendering with temporal coherence and abstraction. In *Proc. NPAR* (2006), pp. 141–149. 8

[BMWD14] BIDAYE S. S., MACHACEK C., WU Y., DICKSON B. J.: Neuronal control of drosophila walking direction. *Science 344*, 6179 (2014), 97–101. 2

[Bra14] Brainbase. Institute of Molecular Pathology Vienna, 2014. http://brainbase.imp.ac.at. 2

[Bre04] BREWER C.: *Designing Better Maps: A Guide for Gis Users*. Environmental Systems Research, 2004. 4

[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. 3

[Car84] CARPENTER L.: The a-buffer, an antialiased hidden surface method. In *Proc. SIGGRAPH* (1984), pp. 103–108. 5

[CLea11] CHIANG A.-S., LIN C.-Y., ET AL.: Three-dimensional reconstruction of brain-wide wiring networks in drosophila at single-cell resolution. *Current Biology 21*, 1 (2011), 1–11. 2

[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. 6

[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. 3

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

[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. 1

[ISea14] ITO K., SHINOMIYA K., ET AL: A systematic nomenclature for the insect brain. *Neuron 81*, 4 (2014), 755–765. 3

[Itt03] ITTEN J.: *Kunst der Farbe. Studienausgabe. Subjektives Erleben und objektives Erkennen als Wege zur Kunst.* Urania, Stuttgart, 2003. 4

[JRea12] JENETT A. A., RUBIN G. M. G., ET AL.: A gal4-driver line resource for drosophila neurobiology. *Cell Reports 2*, 4 (2012), 991–1001. 2

[KEBT10] KLANTEN R., EHMANN S., BOURQUIN N., TISSOT T.: *Data Flow 2 - Visualizing Information in Graphic Design*. Gestalten Verlag, 2010. 4

[KMS07] KALKOFEN D., MENDEZ E., SCHMALSTIEG D.: Interactive focus and context visualization for augmented reality. In *Proc. ISMAR* (2007), pp. 1–10. 5

[KW05] KLANT M., WALCH J.: *Grundkurs Kunst, Band 1-3*. Schroedel Verlag, 2002,2003,2005. 4

[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. 2

[LR93] LEWIS C., RIEMAN J.: Task-centered user interface design: A practical introduction. a shareware book published by the authors, 1993. 9

[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. *Proc. IEEE PacificVis* (2011), 35–42. 2, 3

[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. 2

[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. 1

[PPW91] PETERS A., PALAY S. L., WEBSTER H. D.: The fine structure of the nervous system: Neurons and their supporting cells. *Oxford University Press* (1991). 2

[SBS*13] SORGER J., BÜHLER K., SCHULZE F., LIU T., DICKSON B.: neuromap - interactive graph-visualization of the fruit fly's neural circuit. In *Proc. IEEE BioVis* (2013). 2, 3

[Sim09] SIMPSON J. H.: Mapping and manipulating neural circuits in the fly brain. *Advances in genetics 65* (2009), 79–143. 1

[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. 2

[Spe07] SPENCE R.: *Information Visualization: Design for Interaction*. Prentice Hall, 2007. 4

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-d shapes. In *Proc. SIGGRAPH* (1990), pp. 197–206. 8

[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. 2

[vPvO13] VAN PELT J., VAN OOYEN A.: Estimating neuronal connectivity from axonal and dendritic density fields. *Frontiers in computational neuroscience 7* (2013), 160–160. 2

[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. 2