An Interactive Visualization Tool for Multi-channel Confocal Microscopy Data in Neurobiology Research

Yong Wan*, Hideo Otsuna*, Chi-Bin Chien, and Charles Hansen, *Senior Member, IEEE*

Abstract—Confocal microscopy is widely used in neurobiology for studying the three-dimensional structure of the nervous system. Confocal image data are often multi-channel, with each channel resulting from a different fluorescent dye or fluorescent protein; one channel may have dense data, while another has sparse; and there are often structures at several spatial scales: subneuronal domains, neurons, and large groups of neurons (brain regions). Even qualitative analysis can therefore require visualization using techniques and parameters fine-tuned to a particular dataset. Despite the plethora of volume rendering techniques that have been available for many years, the techniques standardly used in neurobiological research are somewhat rudimentary, such as looking at image slices or maximal intensity projections. Thus there is a real demand from neurobiologists, and biologists in general, for a flexible visualization tool that allows interactive visualization of multi-channel confocal data, with rapid fine-tuning of parameters to reveal the three-dimensional relationships of structures of interest. Together with neurobiologists, we have designed such a tool, choosing visualization methods to suit the characteristics of confocal data and a typical biologist's workflow. We use interactive volume rendering with intuitive settings for multidimensional transfer functions, multiple render modes and multi-views for multi-channel volume data, and embedding of polygon data into volume data for rendering and editing. As an example, we apply this tool to visualize confocal microscopy datasets of the developing zebrafish visual system.

Index Terms—Visualization, neurobiology, confocal microscopy, qualitative analysis, volume rendering.

---◆---

1 INTRODUCTION

There has been a tremendous explosion in the popularity of confocal microscopy [6] in recent years, due to its ability to produce high-quality 3D images, scan fluorescent specimens that have a thickness of hundreds of microns, and generate time sequence images of living cells and tissues as 4D data. The discovery of fluorescent proteins [17] provides an invaluable approach for marking biological targets. When fluorescent proteins or dyes of different emission wave lengths are used for marking different cell/tissue types in confocal scannings, the resulting image datasets are multi-channel.

In neurobiology, confocal technology is widely used for studying the three-dimensional structure of the nervous system; Figure 1 shows the typical workflow. Visualization tools are required for qualitative analysis, which gives an overall evaluation of the experiment results, and higher quality and interactivity of these tools can help researchers decide which quantitative measurements to make, and extract biologically significant conclusions.

However, most neurobiologists' tools for qualitative analysis are rudimentary, such as looking at image slices or maximal intensity projections. There are several academic and commercial visualization packages available, but these have various significant feature limitations when applied to multi-channel confocal data. Despite the plethora of volume rendering techniques that have been available for many years, there is a real demand from neurobiologists, and biologists in general, for a flexible visualization tool that allows interactive visualization of multi-channel confocal data, with rapid fine-tuning of parameters to reveal the three-dimensional relationships of structures

---

- *Yong Wan* is with Scientific and Imaging Institute at University of Utah, E-mail: wanyong@cs.utah.edu.*
- *Hideo Otsuna* is with Department of Neurobiology and Anatomy at University of Utah, E-mail: ostuna@neuro.utah.edu.*
- *Chi-Bin Chien is with Department of Neurobiology and Anatomy at University of Utah, E-mail: chi-bin.chien@neuro.utah.edu.*
- *Charles Hansen is with Scientific and Imaging Institute at University of Utah, E-mail: hansen@cs.utah.edu.*
- ** These authors contributed equally to this work.*

of interest.

Confocal microscopy data have their own characteristics, which differ from other biomedical data, such as CT or MRI, which must be taken into consideration as we design such a tool for confocal microscopy visualization:

Multi-channel data: As mentioned above, labeling with different fluorescent proteins and fluorescent dyes yields multi-channel data, with each channel representing a different cell or tissue type. Usually the data in different channels are spatially interwoven, with data from one channel having the highest interest, such as the channel containing labeled neuron fibers.

Subtle boundaries: Clearly visualized boundaries of brain regions are often essential for analysis, as when analyzing connectivity of neuron fibers between regions [15, 21]. However, biologically meaningful boundaries may be only subtly presented in the confocal data, and may be present in only one channel of the multi-channel data. Thus, boundary segmentation must often be done manually.

Finely detailed structures: Biomedical techniques such as antibody staining and gene transfer allow delivery of fluorescent dyes to specific cell or tissue types, which can result in very finely detailed structures, such as neuronal fibers or synapses.

Visual occluders and noise: Structures irrelevant to the analysis may also be labeled through the fluorescent staining process, resulting in visual occluders that obscure the structures to be visualized. Fine detailed structures can also be obscured by noisy data, due to statistical noise or electronic noise from the scanning device [7].

Working together with neurobiologists, we have designed an interactive visualization tool, which suits a typical biologist's workflow and meets the challenges listed above. The contributions of our work and this application paper to visualizing confocal microscopy data are:

Interactive settings of volume rendering properties to maximize rendering quality: For better rendering quality and depth perception, we add shading and depth cueing to volume rendering. For detail enhancement and noise suppression, a 2D transfer function can be set through intuitive parameters. All the volume rendering parameters take effect interactively.

Multi-modes and multi-views for multi-channel data visualization: The multi-channel dataset can be combined in a single render view with different render modes, with each mode showing a different aspect of the data. With multi-view, different render modes can be displayed at the same time, or several datasets can be compared.

Embedding polygon data into volume data for region definition and volume editing: Biological boundaries are usually manually extracted as polygon data with segmentation tools. These polygon data can be rendered together with volume data, which is a clear and efficient way to show the regions of interest. Furthermore, polygon data can be used to trim the volume data, and volume data within different regions defined by polygon data can have different property settings to aid visualization.
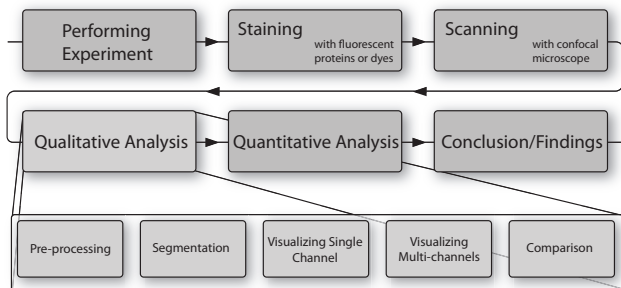

Fig. 1. A typical workflow in neurobiology research with confocal microscopy.

2 RELATED WORK

We have drawn our techniques from previous work on 3D visualization, including volume rendering, transfer function settings, and polygon rendering.

Cai and Sakas [4] proposed three levels of data intermixing and rendering pipelines in direct multi-volume rendering, which include image level intensity intermixing, accumulation level opacity intermixing, and illumination model level parameter intermixing. They applied their method to radiotherapy treatment planning, and compared the features of each method. Rossler et al. [20] described a framework for GPU-based multi-volume rendering, which was used for the visualization of functional brain images. In their framework, they provide a correct overlaying of an arbitrary number of volumes, with the visual output for each volume independently controlled. In his thesis work, Grimm [8] presented a full-blown high-quality raycasting system, which can efficiently process and visualize multiple large medical volume datasets.

Kniss et al. [11] proposed using multidimensional transfer functions for interactive volume rendering, and used a set of direct manipulation widgets for transfer function settings. Seg3D [22] uses the widgets described in Kniss's paper to set 2D transfer functions for volume rendering. Rezk-Salama et al. [19] presented a framework for implementing semantic models for transfer function assignment in volume rendering applications and demonstrated that semantic models can effectively be used to hide the complexity of visual parameter assignment from the non-expert user for a specific examination purpose.

Everitt [5] described an algorithm for interactively rendering order-independent transparent polygon objects, also known as depth peeling, with graphics hardware. The depth peeling algorithm is widely used for correctly blending transparent polygon meshes. Kreeger and Kaufman [12] presented an algorithm that embeds opaque and/or translucent polygons within volumetric data, by rendering thin slabs of the translucent polygons between volume slices using slice-order volume rendering. They demonstrated their algorithm with examples of medical applications and flight simulators. Nagy and Klein [14] presented the concept of volumetric depth-peeling, and they separated the volume data into interior and exterior based on a fixed iso-value. Weiskopf et al. [26] proposed clipping methods that are capable of using complex geometries for volume clipping, which enable selecting and exploring subregions of the dataset.

There is excellent previous work on visualization and segmentation of data from optical microscopes. Janoos et al. [10] presented a method to reconstruct dendrites and spines from optical microscope data by using a surface representation, and the dendrites and spines are visualized in a manner that displays the spines' types and the inherent uncertainty in identification and classification. Mosaliganti et al. [13] described methods to reconstruct cellular biological structures from optical microscopy data, and they applied their methods to light, confocal and phase-contrast microscopy data.

There are some commercially available software packages that can be used for visualizing confocal data. Amira [25] can render volume datasets from confocal microscopes, and visualize them together with polygon data, which are usually generated by its segmentation tool automatically or manually. Imaris [2] incorporates multiple volume rendering algorithms for visualizing microscopy data interactively, and it can also generate polygon data for rendering or volume editing. Volocity [9] can load multi-channel confocal data, and it provides both interactive and non-interactive volume renderers for visualizing them. The neurobiologist users often feel there are still problems with these tools: many don't provide adequate parameter settings for fine-tuning volume rendering results; some are not interactive when adjusting parameters; and it is always laborious to analyze repetitive experiments.

3 VISUALIZATION OF CONFOCAL MICROSCOPY DATA FOR QUALITATIVE ANALYSIS

3.1 A Workflow of Qualitative Analysis of Confocal Microscopy Data

Figure 1 shows a detailed workflow of qualitative analysis, which in neurobiological research, answers questions such as whether certain types of cells are present in a region, how neuron fibers connect different regions, and if there is a difference between samples. In this workflow, pre-processing often consists of basic image processing techniques such as noise reduction and contrast enhancement; median filters are usually used for noise reduction [16]. Segmentation and visualization steps are sometimes iterative, involving generation of polygon data, combining the rendering of polygon data and volume data, and regenerating polygon data. In the visualization steps, details of the datasets are examined, which requires fine-tuned rendering quality with great interactivity. In the comparison step, datasets from different samples are often compared, such as datasets of a mutant and a wildtype sample of zebrafish, or datasets from replicate experiments.

3.2 Interactive Volume Rendering and Rendering Quality Enhancement

We use GPU slice-based volume rendering for real-time display and user interaction. Optical properties, color information, and opacities are assigned and blended [24]. Compared to looking at image slices and maximum intensity projections, our tool can provide better perception of the spatial structures. Compared to volume rendering methods previously used in neurobiology, our tool has the advantage of providing a strong visual cue for orientation and depth, and high interactivity. It is useful not only for single dataset visualization but for comparing several different samples, especially when the datasets are scanned with samples oriented differently.

3.2.1 Shading and Depth Cueing

Our tool provides better perception for 3D spatial structures by adding shading and depth cueing to the volume rendering [23, 3]. User adjustable settings allow fine-tuning of these effects.

Shading is calculated according to the Phong model [18], where normals are approximated from gradients and stored in the 3D textures. Figure 2A shows the default shading effect, and Figure 2B shows the result after changing the ambient intensity, and shapes of local features, such as individual cells, are better perceived.
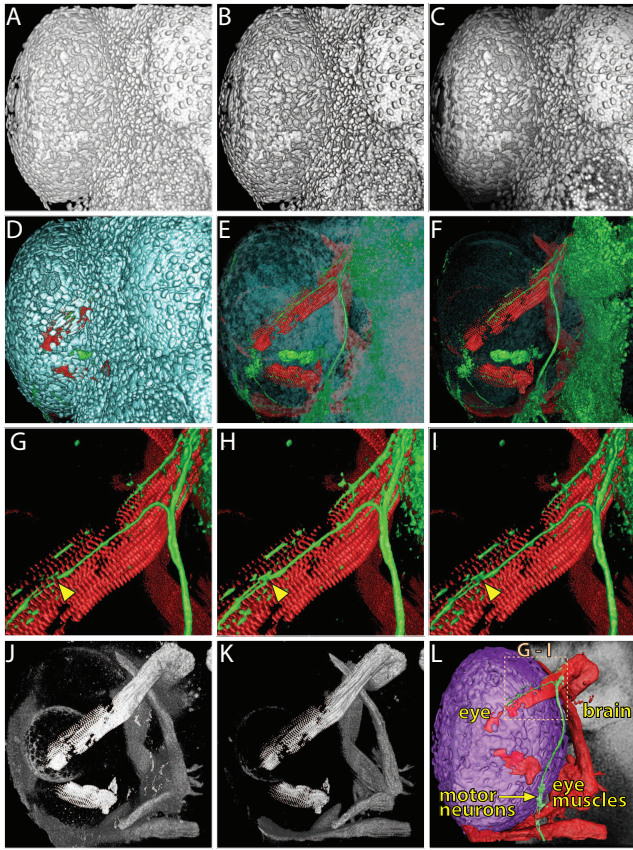
Fig. 2. Rendering effects. A: Default shading effect; B: Lowering the ambient intensity increases the contrast for local features; C: Depth cueing can better show the overall shape; D: Cyan colored channel (all cell nuclei) obstructs other channels; E: Increasing the transparency may be helpful, but it makes the rendering obscure, and underlying channels are still partially occluded; F: Increasing the boundary extraction value can better show the spreading of the cells and underlying channels; G: The motor neurons (green) projecting to the eye muscles appear artifactually disconnected (arrowhead); H: Adjusting the offset value reveals that motor neuron fibers are in fact connected; I: Shading helps better define the shape; J: Noise is superimposed on the data of interest in the red channel (eye muscles); K: Increasing the low threshold suppresses the noise; L: A map of the regions analyzed. (Dataset: Zebrafish head)

Our tool lets user set the voxel aspect ratio for loaded volume datasets, as confocal volume usually has lower Z resolution and thus the voxels are anisotropic. To avoid lighting artifacts caused by changing voxel aspect ratio, the pre-calculated normals are rescaled in the shader programs according to user settings.

Depth cueing is applied by attenuating the intensity values according to the relative depths of voxels, and the attenuated intensity value is calculated with the following equation:

$$V_{attn} = f \times V_{bg} + (1 - f) \times V_{data}, f = \frac{d_{data} - d_{front}}{d_{back} - d_{front}} \times V_{scale}$$

Where V_{attn} is the attenuated intensity, V_{bg} is the background intensity, V_{data} is the voxel intensity, d_{data}, d_{front} and d_{back} are the distances of the voxel, front and back of data from the view point, and V_{scale} is a parameter controlling how much attenuation is applied, which can be adjusted by the user. Figure 2C shows the results after depth cueing is applied, where the overall shape of the 3D structure is more apparent than with only shading (Figure 2B).

3.2.2 Intuitive and Efficient Transfer Function Settings

2D transfer functions [11] are used for setting rendering properties of volume data, as their boundary extracting capability can render fine structures from confocal data. We found, however, that neurobiolo-

gists prefer intuitiveness and efficiency to complicated transfer function widgets and settings. With this in mind, we chose a family of the 2D transfer functions that best suits confocal data structure extraction, while the parameters for fine-tuning the shapes of the transfer function are chosen and named for better operability. The shape of the 2D transfer function, as well as the parameters, are illustrated in Figure 3.
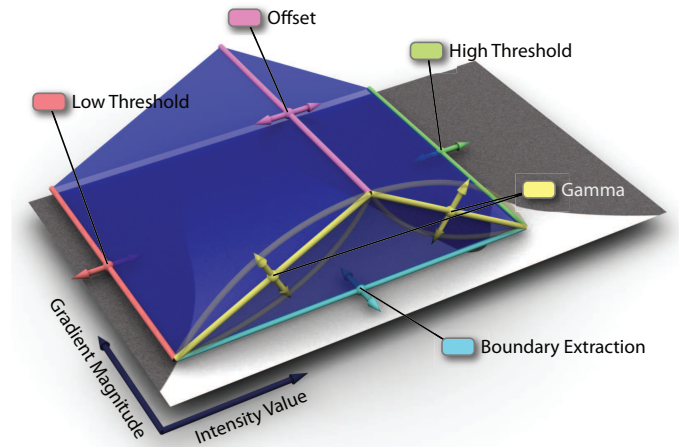

Fig. 3. 2D Transfer function and its parameters.

Figure 3 shows a joint histogram of the volume data, its axes being intensity value and gradient magnitude. The 2D transfer function occupies a rectangular region of the histogram and has a tent-like shape. The meanings of the parameters are:

Boundary extraction: Controls the cut-off value of gradient magnitude. Setting a higher value can isolate better-defined boundaries in the volume data. Figure 2F shows that spreading of nuclei is seen in a combined rendering with other channels. By increasing the boundary extraction value, only the voxels defining nucleus boundaries are rendered. Combined with transparency adjustment, both the underlying channels and the spreading of nuclei are seen, which is not possible by adjusting transparency solely (Figure 2E).

Offset: Sets the intensity peak in the 2D transfer function, so that voxels with the corresponding intensity value are accentuated. Figure 2H and I show that the continuity of neuron fibers is recovered after adjusting intensity offset.

Low and high thresholds: Set the low and high cut-off values of scalar intensity. These values are useful for noise suppression. Figure 2J and K show an example before and after the threshold values are adjusted; noisy data are eliminated after adjusting the low threshold value.

Gamma: Controls how values off the intensity peak are attenuated by adjusting the exponent of the intensity values. Gamma is adjusted to get a better contrast of the output renderings.

For multi-channel volume dataset, transfer function for each channel can be adjusted independently. Our tool lets users interact with a limited set of parameters, with each parameter adjusted by either linked slider or numerical entry. The corresponding parameter settings in the user interface are listed in Figure 9. By avoiding complicated widgets or the jargon of transfer function settings, the provided interface is more intuitive for neurobiologists to use and can quickly obtain the desired visualization results. Users can also save the settings of previous work, and apply them to similar datasets, or use them as a starting point for later fine-tuning, which further accelerates the visualization workflow.
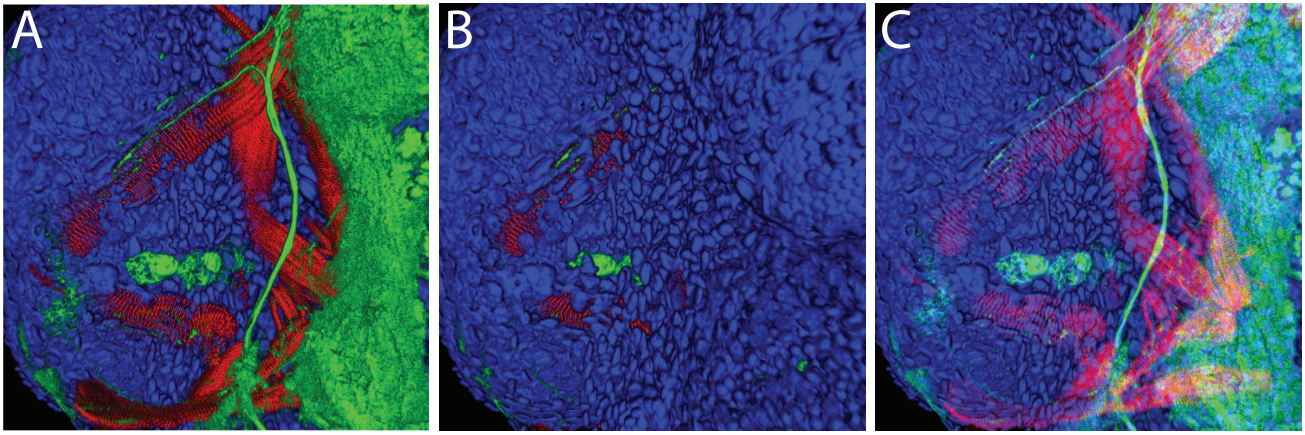
Fig. 4. Render modes. A: Layered; B: Depth sorted; C: Composite. In layered mode (A), the rendering order of the channels, from top to bottom, is: neurons (green), muscles (red), and all cell nuclei (blue). The fibers of motor neurons can be observed without any obstruction, even when user rotates the view. In depth sorted mode (B), almost all the information from other channels is covered by that of all cell nuclei (blue). Increasing the transparency of the obstructing channel makes it obscure and less detailed, as seen in Figure 2E. With composite mode (C), all the channels can be seen at the same time, as well as the fine details. (Dataset: Zebrafish head)

3.3 Multi-modes and Multi-views for Multi-channel Volume Data

For multi-channel confocal microscopy data, qualitative analysis usually requires visualizing the spatial relationship between data from different channels. When combined together, however, data from different channels often interfere with each other, and details of interest from one channel can be occluded. Our tool provides three render modes suggested by our collaborating neurobiologists for multi-channel volume data. When used jointly, both the spatial relationships and details can be visualized clearly. Figure 4 compares the results of same three-channel dataset with different modes, and the three render modes are:

Layered mode: Similar to layers in 2D painting software, the volume data are layered on top of one another, rendered in the order of channels specified by the user. In this mode, the top layer data cover the lower ones. This does not respect the relative depth relationships within the data, especially during user interaction. Visualization experts did not expect this mode to be effective. Surprisingly neurobiologists often prefer this mode since it can better show fine inner structures, such as neuron fibers, when placed in the top layer (Figure 4A).

Depth sorted mode: The multi-channel volume data are blended first for each polygon slice and then the slices are blended together. This is the correct way to show the spatial relationships between channels, and most visualization tools that support multi-channel datasets use this mode. But sometimes the fine structures from one channel are covered by voxels from other channels with lower depth values. Lowering the transparency of the obstructing data can reveal the deeper structures, but usually the details of the obstructing data are lost (Figure 4B).

Composite mode: This is the image level intermixing described by Cai and Sakas [4]. Each dataset of the multi-channel volume data is first rendered into a texture, and the textures are composed into the final rendering with color component addition. As shown in Figure 4C, information from all channels can be seen at the same time, as long as distinguishable colors are used. As it is not necessary to increase the transparency of the occluding channels, the renderings of all channels are bright and full of details. As most datasets in confocal research have three channels or less, it is most effective to set colors as pure red, green, and blue. Shading effect calculation is clamped to single color components if data channels are set to pure red, green, and blue. So the original data

channel information can still be extracted from the exported images of this mode, by separating the color channels. And this is important for further processing and publishing.

Neurobiologists may find features they need in each mode, with each mode best suiting certain applications. Joint views of different render modes can allow even better data comprehension. We provide an interface to allow the neurobiologists to switch between the render modes quickly, and multiple viewports can be set for different render modes, which can be operated separately, or synchronized to the same viewing direction.

Multi-views are indispensable when comparing different datasets in the qualitative analysis workflow. Datasets from replicate samples or from mutants and wildtypes are visualized and compared in different views. Like the transfer function settings, users can set the views quickly and accurately, or let the tool remember the view settings for later comparison.

3.4 Embedding Polygon Data for Region Definition and Volume Editing

As mentioned above, incorporation of biologically meaningful boundaries can greatly aid interpretation of confocal data. However, boundaries often cannot be reconstructed simply by setting transfer functions or through automatic segmentation, so that polygon data resulting from manual segmentation of the volume data are necessary to visualize the boundaries. For some applications such as crude region definition or volume culling, simple polygon geometries can be generated on the fly, and translated, rotated, or scaled to specific positions. This is less time-consuming than manual segmentation, but is still sufficient for many cases in qualitative analysis where precision is not a major concern.

We use the depth peeling [5] algorithm to solve the ordering problem when multiple transparent objects as well as volume data are rendered. The user can set its accuracy by adjusting the number of peeling layers.

In many applications of qualitative analysis, one peeling layer can achieve a satisfactory result while maintaining high interactivity. With a higher peeling layer setting, better accuracy can be achieved, allowing better understanding of how the volume data and the polygon-defined regions are spatially related. For most complex geometries resulting from confocal data segmentation, we found four layers enough for sufficient accuracy. Figure 5 illustrates the algorithm and Figure 6 compares the difference between the depth peeling settings. The examples show how the positions of neurons relative to the eye and central brain can be better perceived.
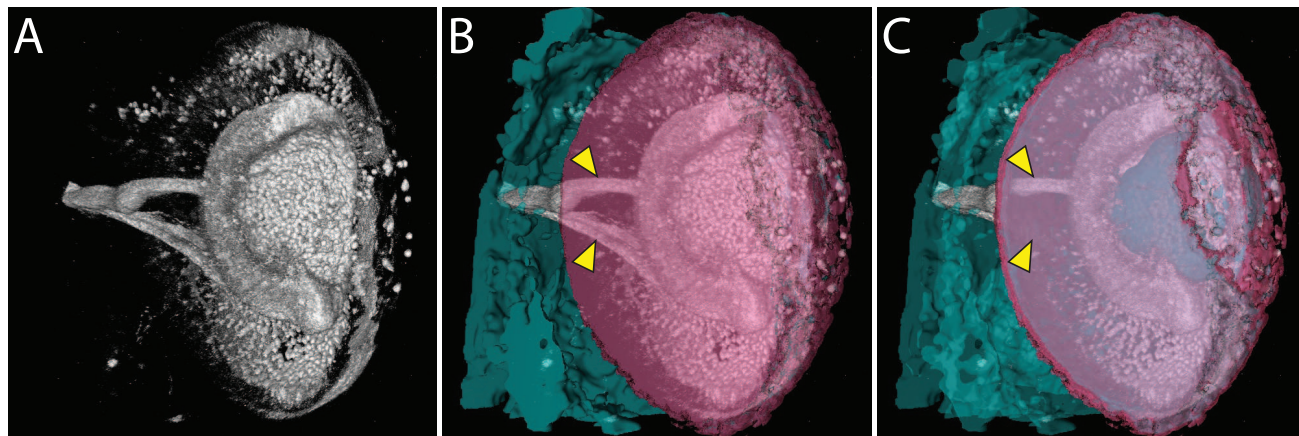
Fig. 6. Depth peeling results. A: Ventral view of the volume data showing retinal ganglion cells connecting between the eye and the brain; B: Polygon data added, separating volume data into eye (magenta) and brain (cyan); depth peeling layers set to one; C: Same data, depth peeling layers set to four. Arrowheads point to two branches of visual neuron fibers. With more depth peeling layers (C), it is clear that the lower branch is located deeper behind the eye region, which is not apparent in either A or B. (Dataset: Zebrafish head)
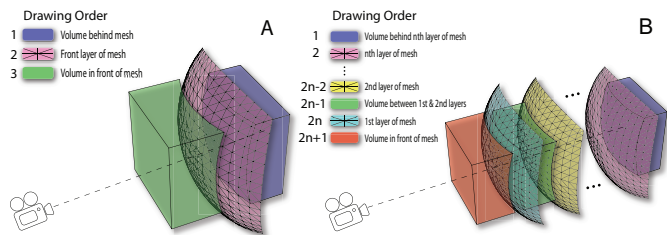

Fig. 5. Depth peeling algorithm. A: Only one depth peeling layer; B: n depth peeling layers.

3.4.1 Volume Editing with Polygon Data

Some visual occluders are clustered and large, and therefore hard to eliminate by using transfer functions. Polygon data can be used to cull these data. We generate voxelized objects [26] from polygon data, which are 3D textures containing information whether a voxel is inside or outside the enclosure defined by polygon data. The mask volume separates data volume into interior and exterior, either of which can be culled. Different channels can share one mask volume, or in most cases, they are culled with different mask volumes. Figure 7 shows how volume culling is applied to a three-channel dataset, where eye muscles and neurons are clearly visualized, and their spatial relationships better revealed, after culling the visual occluders.

Furthermore, different transfer functions can be applied to volume data within different regions defined by polygon data. Figure 8 shows how this is applied to the data from the visual system, where interconnected neurons are color-coded according to the regions where they are situated.

4 RESULTS AND DISCUSSION

4.1 Implementation and Application

In collaboration with neurobiologists, we have realized our design goals and chosen techniques as a working tool, which can aid neurobiologists for qualitative analysis of confocal data. Our implementation uses an OpenGL-based volume rendering library we developed. The input formats of our tool are tiff and nrrd, which are commonly used in medical researches and can be easily converted from manufacturer specific raw formats of confocal microscopes. The tool reads in the volume datasets, pre-calculates gradient fields and 2D histograms. The in-memory datasets are broken into blocks that each can fit into the graphics memory, and the data blocks are sent to the graphics card for rendering as OpenGL texture objects. Shading and depth cueing

effects, 2D transfer function lookup, the render modes, and depth peeling are all coded with OpenGL Shading Language (GLSL). A screenshot of the tool is shown in Figure 9, and its functions and operations are demonstrated in the supplementary video.

As an example, our collaborating neurobiologists applied our tool to visualize Tg(brn3a-hsp70:GFP) transgenic zebrafish embryos (Figure 10), recently described in the neurobiological literature [1, 21], and compared the result with that of using maximum intensity projections. These transgenics express GFP in retinal axons as well as tectal neurons. Our tool's visualizations (Figure 10A-C) illuminated several features that were previously obscured in maximum intensity projections (Figure 10D-F). First, there was a clear boundary between the optic tectum, where the retinal axons terminate, and the cell bodies of the tectal neurons (Figure 10B); this boundary is obscured in the maximum intensity projection (Figure 10E). Second, 3D relationships that are hidden in the maximum intensity projection (Figure 10E) become clear when volume-rendered: the eye and the tectobulbar tract are located deeper than the optic tectum (Figure 10B). Third, volume rendering reveals surface texture (Figure 10C) obscured by pixel saturation in maximum intensity projections (Figure 10F); showing for instance the presence of an arborization field contacted by the retinal axons just before they reach the optic tectum.

4.2 Performance and Rendering Quality Comparison

The tool has been tested and compared to other available packages by neurobiologists, and they found it better suits their research needs in terms of interactivity, rendering quality, and efficiency. Figures 2, 4, 6, 7, 8, 10, and 11 were all generated by neurobiologists in their studies of zebrafish. Table 1 shows the rendering speed of our tool for the two datasets we used in the paper (on Windows PC with Core 2 Quad Q9550 2.83GHz, GeForce GTX 280, 4GB RAM, and 1600x1200 display).

Table 2 compares the operating time of our tool and other commonly used commercial packages. The timings were calculated by studying the video captures of the operations by a fluent user in neurobiology confocal research. The dataset used for comparing is the zebrafish head dataset. In Table 2, data loading is the time from opening the datasets to when they are displayed in the viewport; parameter adjusting is the time used to adjust rendering properties to get satisfying results. Please notice that the user we studied here had worked with the datasets and the tools for quite long time, and was very familiar with the processes. So the timings only reflect the fastest operations possible. Even our tool has more parameters, neurobiologist users find them necessary and easy to work with.

Figure 11 compares the rendering results of same dataset with different tools. The results were generated by a neurobiologist working
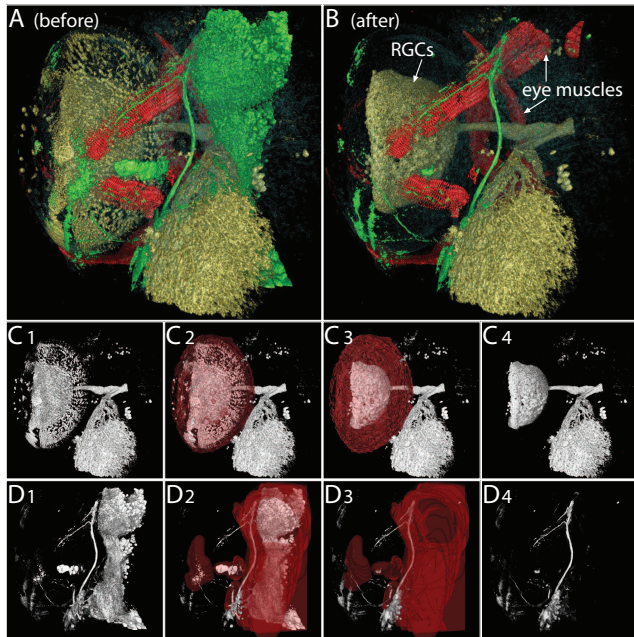
Fig. 7. Volume culling with polygon data. A: Original volume data; B: Volume data after culling; C: The process of culling the occluding data in the yellow channel, showing retinal ganglion cells (RGCs; C1: Original volume data; C2: Polygon data enclosing the layer of photoreceptor cells are loaded; C3: Volume data inside of the region are culled; C4: The volume data after culling); D: Culling the visual occluders in the green channel, showing motor neurons (D1: Original volume data; D2: Polygon data enclosing neuron clusters of the brain are loaded; D3: The volume data inside of the polygon data are culled; D4: The output data show only motor neurons.). (Dataset: Zebrafish head)
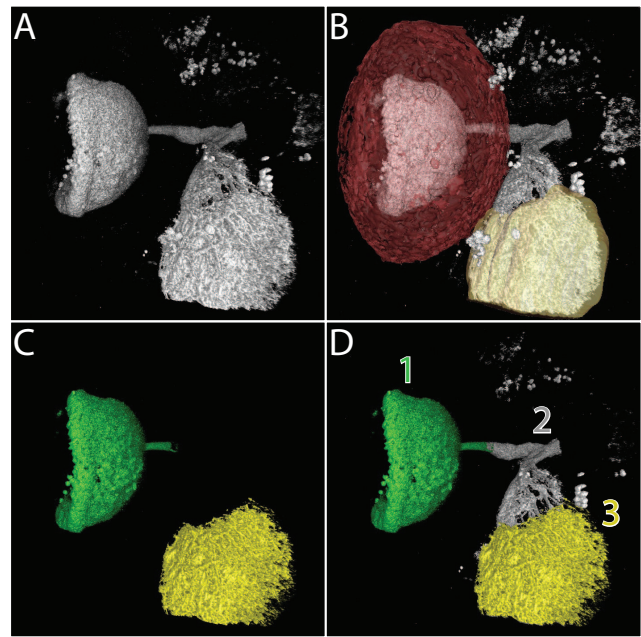

Fig. 8. Different transfer function settings of volume data in different regions. The process shows how cells in different biologically meaningful regions are marked out, where the color-coded volume data represent cells in eye (green) and tectum (yellow) regions respectively. A: The loaded volume data show the eye and tectum; B: Two polygon datasets are loaded, defining the regions of the eye and tectum; C: Connecting neuron fibers between the eye and tectum can be culled; D: Different colors can be set for different regions. (Dataset: Zebrafish head)

with confocal data for eleven years and familiar with all the tools compared. The rendering parameters of each tool were adjusted with the aim of showing details of fine structures as well as overall surface shapes of the sample studied. By rendering the same dataset, we can see each tool's strength and shortcomings in rendering quality. For example, Volocity is good at rendering details of the fibers and cells but doesn't interpret the surface shape as good as Imaris. Our tool can render both local details and global shapes clearly.

4.3 Feature Discussion

Our collaborating neurobiologists compared the tool to other tools that use different rendering techniques, and they concluded that our tool has the best interaction speed, without apparent rendering quality loss. The advantage of a GPU-based volume renderer is most obvious when there are many finely detailed structures in a dataset - usually the user wants to explore the dataset quickly, and often wants to keep high rendering quality during viewport interaction, so that he or she can keep track of certain structural details. Our collaborating neurobiolo-

Table 2. Operating Time Comparison

	Data Loading (sec)	Parameter Adjusting (sec)	Parameters per Channel
Amira 5	80	130	3
Imaris 6	70	180	5
Volocity 4	80	70	3
Our Tool	20	75	11

gists also appreciate the instant visual feedback of rendering parameter changes that a GPU-based volume renderer can provide, where they can quickly fine-tune the rendering properties without waiting for the changes to take effect.

Our collaborating neurobiologists mentioned many times that the available volume rendering packages are not efficient for confocal data. This is because many volume rendering packages try to provide comprehensive settings for volume properties such as transfer function editors, which are sometimes confusing and laborious to work with. In contrast, we analyzed the specific features of confocal data, and designed parameter settings accordingly. The neurobiologists found the set of parameters we provided for transfer function manipulation are more intuitive for confocal data; they can often get the desired results within minutes. This is especially important when there are multiple datasets to process as in high-throughput microscopy.

As mentioned in the introductory section, most confocal datasets are multi-channel. Our collaborating neurobiologists felt that many available tools either neglected this important feature completely, or did not pay much attention as to how to present different channels together, yet render clearly both individual channels and the relationships between them. They found the multiple render modes of our tool a good way to handle multi-channel data. They often start with layered mode, and setting the channel with the finest detail as the top layer. For instance, motor neuron labeling (Figure 4A) has many fibers

Table 1. Rendering Speed of Our Tool

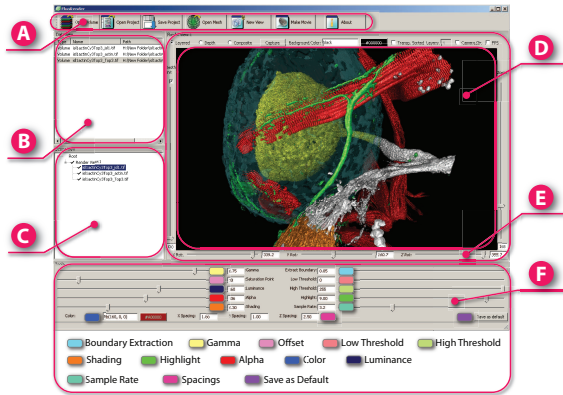|	Zebrafish Head Dataset	Zebrafish Visual System Dataset
X Res.	512	512
Y Res.	512	512
Z Res.	160	136
Data Length	8 bit	8 bit
Samples/Voxel	3.2	3.2
Channels	3	2
FPS	12.3	24.5

Fig. 9. A screenshot of our tool. A: Toolbar, and the buttons are: Open Volume, Open Project, Save Project, Open Mesh, New View, Make Movie, and Info. B: Data View, loaded datasets are listed. C: Scene View, manages the viewports and associated datasets. D: Render View, displays the datasets. E: Viewport Settings, top: render modes, screen capture button, background color setting, depth peeling layers; left: depth attenuation setting; right: zoom factor; bottom: rotation angles. F: Volume Property Settings, items are listed in figure, and items for transfer function settings correspond to those in Figure 3.

and is otherwise easily occluded by other channels. They then switch to other modes (Figure 4B and C) to better perceive spatial relationships. The synchronized multi-views are often used for displaying the different modes at the same time, as their advantages complement each other.

As mentioned in section 3.4, boundary extraction typically needs to be done manually. Thus, our collaborating neurobiologists appreciate the flexibility our tool provides, of loading either manually or automatically segmented polygon data, and of creating and manipulating simple polygon geometries, for volume editing. We also found that using polygon data is probably the easiest method for volume editing, as the process resembles that in a polygon-based 3D modeling tool. By cutting volume data and setting properties for different subregions, our collaborating neurobiologists found they could make more elegant and effective visualizations of their confocal data.

Through our development process, the feature most emphasized on by our collaborating neurobiologists, was not rendering quality, but the user interface. They found that many similar tools are frustrating to use because of their user interface. We studied the workflow and operative behaviors of a typical neurobiologist carrying out his research on confocal data, and accordingly fine-tuned the user interface of our tool. Some small features, such as providing multiple methods for viewport interaction including mouse dragging, slider adjusting and numerical entering, saving frequently used parameters as a user's default, synchronizing the multi-view for comparison, or even laying out the user interface elements at handy positions, are surprisingly highly valued by our collaborating neurobiologists. They found that these features accelerate the workflow greatly, especially for repetitive analysis.

5 CONCLUSION AND FUTURE WORK

In this paper, we have presented an interactive visualization tool for multi-channel confocal microscopy data in neurobiology research. We followed the typical workflow of a neurobiologist, and discussed how visualization techniques, such as interactive volume rendering, shading and depth cueing for volume data, transfer function settings, and embedding polygon data into volume data for region definition and editing, are applied to the qualitative analysis of the datasets. We also explored how to make the workflow easier and more efficient. Available commercial tools were deemed lacking by neurobiologists, leading to the design goals stated in the introductory section, and our neurobiologist coauthors found the new tool allows them to better perform analysis and high-throughput neurobiology studies.
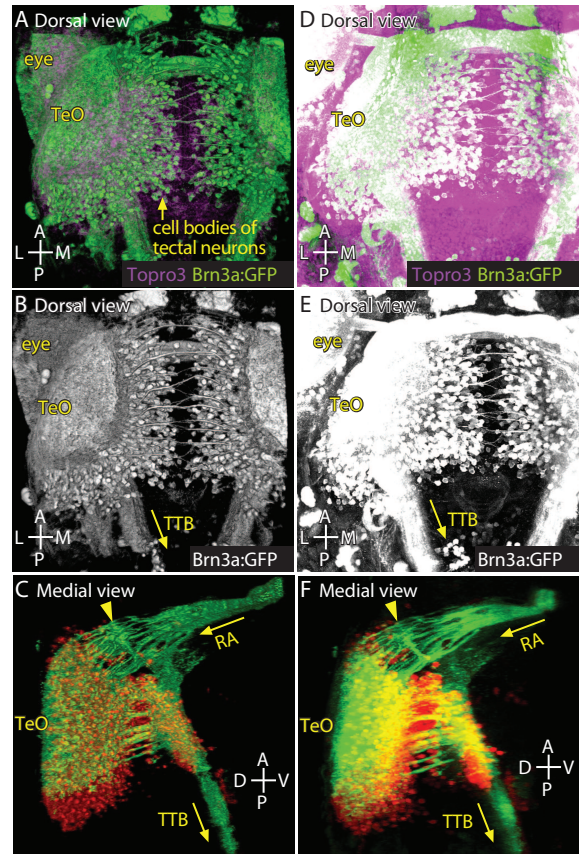

Fig. 10. The zebrafish visual system, rendered with our tool (A-C) and compared to previously used maximum intensity projections (D-F). A, D: Dorsal views of neurons expressing Tg(brn3a-hsp70:GFP) (green), and all nuclei (magenta). B, E: Dorsal views of Tg(brn3a-hsp70:GFP)-expressing neurons only. C, F: Medial view of Tg(brn3a-hsp70:GFP)-expressing neurons. Red, cell bodies colocalized with nuclear staining; green, neural fibers. Arrowhead indicates pretectal arborization field. TeO, optic tectum; TTB, tectobulbar tract; RA, retinal axon. A, anterior; P, posterior; D, dorsal; V, ventral; M, medial; L, lateral. (Dataset: Zebrafish visual system)

For future work, we would like to work on larger datasets and temporal data sequences, as well as the integration of focus-plus-context techniques. Neurobiologists would like to find methods to visualize the temporal development of the volume confocal data in real-time, such as growth of the zebrafish embryo. It would also be of interest to add the most recent research results in volume data segmentation and apply them for both easier segmentation and segmentation of temporal data.

REFERENCES

[1] H. Aizawa, I. Bianco, T. Hamaoka, T. Miyashita, O. Uemura, M. Concha, C. Russell, S. Wilson, and H. Okamoto. Laterotopic representation of left-right information onto the dorso-ventral axis of a zebrafish midbrain target nucleus. *Current Biology*, 15:238–243, 2005.

[2] Bitplane AG. *Imaris*, 2009. http://www.bitplane.com/go/products/imaris.

[3] H. H. Blthoff and H. A. Mallot. Interaction of different modules in depth perception. *IEEE/IAPR First Intl. Conf on Computer Vision*, pages 295–305, 1987.

[4] W. Cai and G. Sakas. Data intermixing and multi-volume rendering. *Computer Graphics Forum*, 18(3):359–368, 1999.

[5] C. Everitt. *Interactive Order-Independent Transparency*. White paper, Nvidia, 1999. http://developer.nvidia.com/object/Interactive_Order_Transparency.html.

[6] T. J. Fellers and M. W. Davidson. *Introduction to Confocal Microscopy*, 2008. http://www.olympusconfocal.com/theory/confocalintro.html.

[7] T. J. Fellers, K. M. Vogt, and M. W. Davidson. *CCD Signal-To-Noise Ratio*, 2008. http://www.microscopyu.com/tutorials/java/digitalimaging/signaltonoise/index.html.

[8] S. Grimm. *Real-Time Mono- and Multi-Volume Rendering of Large Medical Datasets on Standard PC Hardware*. PhD thesis, Vienna University of Technology, Gaullachergasse 33/35, 1160 Vienna, Austria, February 2005.

[9] Improvision. *Volocity, High performance 3D imaging software*, 2008. http://www.improvision.com/products/volocity/visualization/.

[10] F. Janoos, B. Nouansengsy, X. Xu, R. Machiraju, and S. T. Wong. Classification and uncertainty visualization of dendritic spines from optical microscopy imaging. *Computer Graphics Forum*, 27(3):879–886, may 2008.

[11] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

[12] K. Kreeger and A. Kaufman. Mixing translucent polygons with volumes. *In Proceedings of IEEE Visualization 1999*, pages 191–198, 1999.

[13] K. Mosaliganti, L. Cooper, R. Sharp, R. Machiraju, G. Leone, K. Huang, and J. Saltz. Reconstruction of cellular biological structures from optical microscopy data. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):863–876, 2008.

[14] Z. Nagy and R. Klein. Depth-peeling for texture-based volume rendering. *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, pages 429–433, 2003.

[15] H. Otsuna and K. Ito. Systematic analysis of the visual projection neurons of drosophila melanogaster. i. lobula-specific pathways. *Journal of Comparative Neurology*, 497(6):928–958, 2006.

[16] J. B. Pawley. *Handbook of Biological Confocal Microscopy, 2nd edition*. Springer, 1995.

[17] D. W. Piston, G. H. Patterson, J. Lippincott-Schwartz, N. S. Claxton, and M. W. Davidson. *Introduction to Fluorescent Proteins*, 2008. http://www.microscopyu.com/articles/livecellimaging/fpintro.html.

[18] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume rendering on standard pc graphics hardware using multi-textures and multi-stage rasterization. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 109–118, New York, NY, USA, 2000. ACM.

[19] C. Rezk-Salama, M. Keller, and P. Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1021–1028, 2006.

[20] F. Rossler, E. Tejada, T. Fangmeier, T. Ertl, and M. Knauff. Gpu-based multi-volume rendering for the visualization of functional brain images. *In Proceedings of SimVis 2006*, pages 305–318, 2006.

[21] T. Sato, T. Hamaoka, H. Aizawa, T. Hosoya, and H. Okamoto. Genetic single-cell mosaic analysis implicates ephrinb2 reverse signaling in projections from the posterior tectum to the hindbrain in zebrafish. *Journal of Neuroscience*, 27(20):5271–5279, 2007.

[22] SCI Institute, University of Utah. *Seg3D*, 2008. http://software.sci.utah.edu/SCIRunDocs/index.php/CIBC:Seg3D.

[23] T. J. T and M. E. Perception of surface curvature and direction of illumination from patterns of shading. *Journal of Experimental Psychology: Human Perception and Performance*, 9(4):583–595, 1983.

[24] A. VanGelder and K. Kim. Direct volume rendering with shading via three-dimensional textures. In *1996 Volume Visualization Symposium*, pages 23–30. IEEE, 1996.

[25] Visage Imaging. *Amira*, 2008. http://www.amiravis.com/overview.html.

[26] D. Weiskopf, K. Engel, and T. Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):298–312, 2003.
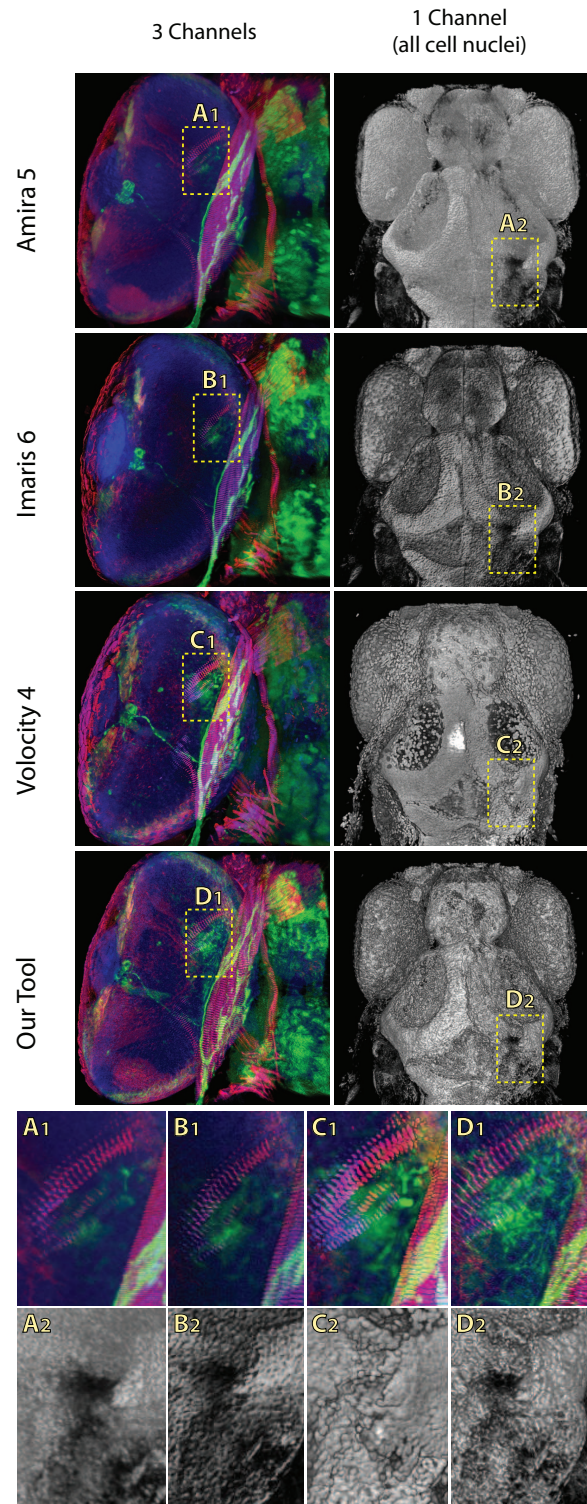
Fig. 11. The rendering result comparison between commercial packages and our tool. Left column: renderings of three channels (red: muscles, green: neurons, blue: all cell nuclei). The dashed rectangular regions (A1-D1) show the fine details of the neurons inside the eye, which can be better seen in our tool. Right column: renderings of single channel (all cell nuclei). The dashed rectangular regions (A2-D2) show a ruptured region of the tissues. The indentation of the damaged tissues can be better observed in our tool. (Dataset: Zebrafish head)