

Interactive 3D Volume Rendering in Biomedical Publications

Bernhard Ruthensteiner^{a*}, Natalie Baeumler^a, David G. Barnes^b

^aBavarian State Collection of Zoology, Münchhausenstraße 21, 81247 München, Germany

^bCentre for Astrophysics and Supercomputing, Swinburne University of Technology, P.O. Box 218, Hawthorn, Victoria 3122, Australia

*Corresponding author at: Bavarian State Collection of Zoology, Münchhausenstraße 21, 81247 München, Germany. Tel.: +49(0)89-8107-135; fax: +49(0)89-8107-300, E-mail address: BRuthensteiner@zsm.mwn.de.

Abstract

We present three examples of interactive, 3D volume rendering models embedded in a PDF publication. The examples are drawn from three different morphological methods—confocal microscopy, serial sectioning and microcomputed tomography—performed on members of the phylum Mollusca. A description of the entire technical procedure from specimen preparation to embedding of the visual model including 3D labels in the document is provided. For comparison, volume rendering with standard visualisation software, and surface rendering incorporated in the 3D PDF figures, are provided. The principal advantages and disadvantages of the techniques and models are discussed. Volume rendering for serial sections is relatively work-intensive, while confocal data have limitations in terms of 3D presentation. Volume renderings are normally downsampled in resolution to achieve a reasonable PDF file size, however intentional information is largely retained. We conclude that volume rendering of 3D data sets is a valuable technique and should become standard in PDF versions of biomedical publications.

Key words: Electronic PDF publication, serial sections, confocal laser scanning microscopy, MicroCT, morphology

Introduction

The capability to display interactive, three-dimensional (3D) models in Adobe Portable Document Format (PDF) files arose in 2005 (PDF version 1.6). The term “interactive” principally refers to the ability of the end-user to view the model from any direction, but can also include the ability to modify certain rendering parameters and control which sections, or components, of a model are displayed. This new illustration technique (hereafter, “3D PDF figure”) was first utilized in academic science publications in 2008 (e.g. Barnes & Fluke, 2008: astronomy; Ruthensteiner & Heß, 2008: life sciences). Both studies suggest a range of possible applications, and present examples. While Ruthensteiner & Heß restrict themselves to surface rendering (SR) models, which are relatively easily created and added to PDF documents using standard tools, Barnes & Fluke embedded novel examples prepared with custom software. Specifically, in applying their own software tool (the S2PLOT library: Barnes et al., 2006), Barnes & Fluke demonstrated a 3D PDF figure of a volume rendering (VR) of a cosmology dataset. Their technique has also been applied to a study on bacterial adhesion to titanium surfaces (Truong et al., 2009).

SR and VR are fundamentally different in the way they transfer data to the visual space. In SR, physical components in a 3D image are displayed as (normally closed) coloured surfaces, often iso-valued surfaces (hereafter, “isosurfaces”). In VR (technical principles extensively reviewed by e.g. Kaufman, A. & Mueller, 2005), the base data cells (voxels, the 3D analogue of pixels) are shown directly with attributes like colour, transparency and luminosity, resulting in a diffuse object with more or less transparent internal structures. While SR provides a more quantitative and local view of a 3D dataset (eg. numerically precise surfaces conceal inner structure), VR yields a global, qualitative view. Accordingly, VR has a different appearance to SR, and can provoke or support new interpretations of the data. VR gained wide acceptance as a computer graphics technique (e.g. Dreben et al. 1988) about a decade after SR (e.g. Fuchs et al., 1977) and has become increasingly

popular in life science publications (e.g. Fishman et al., 2006; Ruffins et al., 2007, Pietsch et al., 2009).

In biomedical science, masses of data with potential of being displayed by VR are acquired at increasing speed. In biology, two methods with results appropriate for VR have become particularly popular: confocal laser scanning microscopy (CLSM) (recent examples: Gruhl, 2009; Evans et al., 2009) and microcomputed tomography (microCT) (recent examples: Ribi et al., 2008; Golding et al., 2009). For both methods virtual section layers are generated, which stands in contrast to histological sectioning or grinding methods, where physical sections are the initial information sources.

VR is routinely applied in the data verification and comprehension phases for data sets gained from methods with virtual sections (e.g. CLSM, microCT). However, we are not aware of this visualization mode being applied to physical section series (e.g. histological sections). The latter are nearly exclusively processed to surface renderings, yet since the basic data—images of sections—are principally the same as those obtained by non-invasive methods, it seems obvious to make use of VR for physical section analysis too.

In this paper, we use three biological datasets of different origins (CLSM, histological section series, microCT) to evaluate the merits and challenges of embedding interactive, 3D VR figures in scientific papers. We compare the techniques of volume and surface visualisation as available in traditional software such as AMIRA, and we demonstrate that these complementary techniques can be combined in 3D PDF figures to enable enhanced comprehension of physical specimens by the readership.

Material & Methods:

Material, specimen preparation and initial graphical treatment:

Example 1 (CLSM, Fig. 1): Adult individuals of *Acroloxus lacustris* were collected south of Vienna (Austria) and maintained in small aquaria where they reproduced. The juvenile specimen (13 days post egg deposition) was freed from the egg capsule and fixed in 4% paraformaldehyde in 0.1m phosphate buffered saline (PBS). After

decalcification in 2% ethylenediaminetetraacetic acid, they were incubated in 0.2% Triton-X in 0.1m PBS and subsequently in 6% normal goat serum (Jackson ImmunoResearch Laboratories, Inc., West Grove, USA) in purified terephthalic acid as blocking reagent. Then the embryos were transferred to the primary antibody, an anti-serotonin (DiaSorin, Stillwater, USA, concentration: 1:800 in blocking reagent, 24 hours at room temperature) and after rinsing in the blocking reagent solution to the TRITC-conjugated secondary antibody (Jackson ImmunoResearch Laboratories, Inc., West Grove, USA, concentration: 1:100 in blocking reagent, 24 hours at room temperature). After dehydration in ethanol, specimens were mounted on slides with an equal mixture of ethanol and 2:1 benzylbenzoate-benzol. Analysis was performed with a Leica DM IRBE CLSM (Leica Microsystems, Wetzlar, Germany) at a wavelength of 514 nm. The image stack (see Table 1 for resolution) was loaded in AMIRA 5.0–5.2.1 (Visage Imaging, Berlin Germany). Here the stack containing the whole specimen was cropped in size (see Table 1) with the *crop editor* to delete areas devoid of nervous tissue signal.

Example 2 (Histological section series, Fig. 2): For all methodical procedures up to segmentation (histology, image treatment, AMIRA alignment) of *Patella caerulea* (Linné, 1758) see Ruthensteiner & Heß (2008, same specimen). The aligned image stack (see Table 1 for resolution)—present in *AmiraMesh* (*.AM) format—was then exported as a set of individual slices saved as TIF-format (Fig. 2E left). Standard image editing software was then used in batch mode to replace black with white (for areas outside of slices prior to alignment), to invert the images, and to enhance the contrast to reduce the background to black (Fig. 2E). Subsequently, the images were individually cleaned with the brush tool to remove artefacts such as section scratches, dirt or brightness gradient, resulting from the histological and photographic procedure. A few images had defects resulting from the histological procedure (e.g. unstained sections), and were manually treated to align their overall brightness and contrast to match the other images. In the resulting negative images the tissue is recorded in different grey values against a black background.

Example 3 (MicroCT data, Fig. 3): The specimen was collected alive, near the marine biological station Observatoire Océanologique de Banyuls-sur-Mer (France) and the shell plates were removed and cleaned from soft tissue by maceration. The plate presented herein has been examined with a μ CT 40 (SCANCO Medical AG,

Brüttsellen, Switzerland) operating at 75 kV. 2D slices of the resulting stack were modified with standard image editing software. First the images were cropped (see Table 1 for resolution). Then, for VR of the internal tubing system, the images were inverted, areas outside the plates were selected with the magic wand tool and stained black, and contrast was enhanced to a degree that only the internal tubings remained as bright areas in black surrounding (Fig. 3A). Surface rendering in AMIRA was performed by threshold segmentation and applying the *fill* function to avoid internal tubing surfaces.

General visualization methods:

Conventional 3D treatment: grey scale TIF-format image stacks were loaded in AMIRA and if required cropped with the *crop editor*. A volume texture module was attached (*Display—Vortex*) and applied using the *color table* mode without maximum intensity projection (*mip*) option in *rgba* and *2D texture* mode. Where downsampling was applied, values (divided by—for x/y/z) are given in the figure legends. The appearance can be manipulated by a variety of parameters—particularly *alpha scale* and *colormap*—adjustments with regard to transparency, shades, gradients, etc., and so the visual emphasis can be changed from internal (Figs 1A,B, 2A) to surface structures (Fig. 2B) or from finer details (Fig. 1B) to more pronounced larger structures (Fig. 1A). Often—for showing internal structures—it makes sense to increase “transparency” by reducing the *alpha scale* value (lower than 1). It should be noted that inversion of images prior to 3D stacking (examples 2, 3) in order to render bright structures in dark surroundings is not required because adjustment of the *colormap* can yield the same result.

Surfaces were generated either as isosurfaces (*Display—Isosurface*) (Figs 1C–E, 2C) or by threshold segmentation (Fig. 3C). Data were *downsampled* for isosurfacing. *Threshold* (brightness value used) was chosen so that the surface matched that of the specimen epidermis. The tessellation of surfaces was simplified with AMIRA and DEEP EXPLORATION (Right Hemisphere, Pleasanton, USA).

Volume renderings as embedded 3D PDF figures (Figs 1D–G, 2D, F–H): TIF-format image stacks were downsampled and converted to greyscale TGA format files using the `IMAGEMAGICK CONVERT` application (available from

<http://www.imagemagick.org>. Downsampling in the x and y directions was necessary at this stage to ensure a manageable PDF file size. In the case of *Acroloxus lacustris* for example, downsampling was applied in the ratio 0.25 : 0.25 : 1 to yield a volume of dimensions 186 x 168 x 139 voxels, stored as 139 TGA files of 186 x 168 pixels (see Table 1 for all examples).

The TGA image stacks were read into our own computer program VOLREN—based on the S2PLOT programming library (Barnes & Fluke, 2006)—which assembled the image stack back into a 3D array, and used S2PLOT's standard volume rendering and isosurfacing (eg. Fig. 1G) functions to create an interactive 3D visualisation of the data. The program code for the version of VOLREN used to create the volume rendering component of Fig. 1 *Acroloxus lacustris* is presented and described in Appendix A.

From VOLREN, standard S2PLOT functionality was used to write the complete visual representation of the rendering to Virtual Reality Modeling Language (VRML, ISO/IEC 14772:1997, <http://www.iso.org>) format. As described in Appendix A, three slice-based volume renderings (one along each of the x, y and z axes) were stored in distinctly-named sections of the VRML “model tree”, *viz.* VRSET1, VRSET2 and VRSET3. Alongside the export of the VRML file (which is simply a text file), the VOLREN program creates (again with standard S2PLOT functionality) a set of images for each of the slice-based volume renderings, that is, a set of yz slices for VRSET1, zx slices for VRSET2 and xy slices for VRSET3. These images are written in TGA format and the IMAGEMAGICK CONVERT application is used to reformat them to PNG format (a bitmap image format supported for import into Acrobat 3D).

Summarising up to this point, we have taken as input a stack of xy slices as TIF-format images, and used the IMAGEMAGICK CONVERT and VOLREN programs to produce output stacks of xy, yz and zx slices as PNG images. As well, we have generated a VRML file that references the three sets of slices. An astute script programmer could in principle achieve this without the S2PLOT library or the VOLREN program. The advantage in using our approach lies in the capacity to produce an interactive volume rendering that is *highly-representative* of the final result that will be rendered in the 3D PDF figure. The user can tune settings such as the opacity mapping (via the `alphafn` discussed in Appendix A), the colour mapping and the relative opacity scaling (via the `ns2sevas` function) to get

precisely the visualisation they want before committing to the export of VRML to target 3D figure creation in PDF.

Final assembling and embedding of 3D PDF figures: volume and surface renderings parts were assembled with 3D REVIEWER, a component of ADOBE ACROBAT PRO EXTENDED version 9 (Adobe Systems Incorporated, San Jose, USA). With this software, labels (*Add 3D Markup* tool), dimensions (*Add Dimension* tool) and the bounding box (*Bounding Box* tool) were added. From 3D REVIEWER, models were subsequently exported to a PDF file. See also Ruthensteiner & Heß (2008) and Kumar et al. (2008) for the procedure of PDF embedding of models. PDF settings were prepared with ADOBE ACROBAT.

Our solution for volume rendering in 3D PDF figures uses a computationally-cheap technique. For a given viewing angle, we select the set of axis-aligned slices (there are three sets: xy, xz and yz) most orthogonal to the viewing angle, and extract and draw these slices from back to front. The slices are drawn as 2D textured facets in 3D space, with colour and opacity blending done by the graphics hardware. This technique is suitable for 3D PDF figures because we have pre-computed and embedded all three texture sets (as PNG image stacks) in the document, and we can select and activate the correct image stack to draw as the user re-orientates the camera around the visualisation. A program code written in Adobe JavaScript 3D (see Appendix B) achieves this effect and is available as a peripheral download on the S2PLOT website. The section of the script that handles the selection of the appropriate image stack as the model is rotated is presented and described in Appendix A. Additionally, by monitoring keyboard and mouse input on the 3D PDF figure, the script also provides most of the basic interactive interface of S2PLOT including features such as ‘a’ for autospin, and ‘+’ and ‘-’ to control zoom.

Results

Example 1: CLSM data (Fig. 1)

CLSM provides for the precise capturing of layers of a total preparation specimen on a microscope equipped with a confocal laser scanning unit (see also Paddock,

1999; Claxton et al., 2008; for physical principles of CLSM). Typically, specimens are stained with a fluorescence dye, which preferentially attaches to selected components. The signal is generated with colour filters specific for the wavelength of the dye. In contrast to standard—also called “wide-field”—epifluorescence microscopy, where there is a true colour light signal excited by the light filter the signal in CLSM purely consists of bright-dark gradient information. The resulting grey value image stack is frequently presented with an artificial (false) colourmap simulating the colour of normal epifluorescence microscopy. For final visualization the image stack is most often processed by software allied with the laser scanning unit into a single maximum-intensity projection (MIP) image; the 3D data is reduced to one or just a few 2D images—often without alterations such as changing the angle by rotating the stack.

The present example shows the serotonergic immunofluorescence stained nervous system of a creeping juvenile (13 days old) freshwater pulmonate gastropod, the lake limpet *Acroloxus lacustris* (Linné, 1758), a common inhabitant of slow-running and stagnant bodies of water across Europe. The result displayed herein shows the components of the nervous system with a concentration of fluorescence signal at sites of ganglionic complexes and fine nervous fibers, of a so-called early creeping stage, a 13 days old juvenile halfway through its development, starting to creep within its egg capsule.

Example 2: Histological section series (Fig. 2)

Biological tissue samples can be sectioned to slices after freezing or after embedding in a variety of media as in the case of “semithin” sectioning. Specimens are embedded in resin and sliced for light microscopy with (mostly) glass or diamond knives. The resulting sections are about 10–20 times as thick (approximately 0.5–4 μm) as so called “(ultra-)thin” sections for electron microscopy. Historically, 3D visualization of physical section series, such as histological section series, is nearly exclusively limited to surface rendering (e.g. Wirkner & Richter, 2008; Baeumler et al., 2008; Schwaha et al., 2008). Typically, substructures are discerned by a manual segmentation procedure followed by surface generation for each component using a standard software algorithm (for general procedures see e.g. Ruthensteiner, 2008; Ruthensteiner & Heß, 2008).

The present example, the metamorphic competent larva of the limpet *Patella caerulea* (Linné, 1758), is the same specimen as has been given in Ruthensteiner & Heß (2008). The results in VR (Figs 2A,B,D,G,H) are similar to the final stack images (Fig. 2E right) utilized, for example, the denser a structure (originally stained more intensely or darker on sections), the brighter it appears in the model. Easily recognizable are features such as cell nuclei, which appear as bright dots. In total, the entire specimen becomes visible in the VR.

Example 3: MicroCT data (Fig. 3)

Microfocus computed tomography (microCT) is computed tomography (CT) optimized for small specimens (for review see e.g. Ritman, 2004; Storck, 2009). CT is a biomedical analysis method that produces 3D volumetric data calculated from a series of X-ray projection images taken at successive rotation angles (often 360° in 1° steps in microCT) (for review see Webb, 2003). Currently, microCT is mainly applied for imaging high contrast-producing structures, such as hard parts of calcareous or chitinous materials. However, methods for discerning different soft tissue components have recently been improved by treatment with contrast enhancing compounds (Ribi et al., 2008; Metscher, 2009). MicroCT data are suitable for being displayed via both VR and SR—depending on the general density and complexity of the 3D data set, or on the physical opacity and compactness of structures to be shown (e.g. Neues & Epple, 2008).

The example shown herein is one of the shell plates of the chiton *Lepidochitona corrugata* (Reeve, 1848), a species that populates the tidal zone of the Mediterranean Sea. Chitons or Polyplacophora represent a rather basal clade of the phylum Mollusca, which dwell in various marine environments. These animals have a creeping sole and a dorsal armour of eight calcareous shell plates. The external surface of the shell plates is endowed with sensory organs, which are supplied from nerves running through an internal tubing system of the shell plates (see e.g. Fernandez et al., 2007). Our results show the internal tubing system of a single shell plate by VR (Fig. 3C,D,F,G) and the external surface by SR (Fig. 3E).

Discussion

Evaluation of present examples

Example 1 (Fig. 1): Immunofluorescence-stained nervous systems acquired by CLSM are chiefly visualized via VR. VR is particularly suitable to this example, as the organ system of interest largely consists of extremely delicate components. The fluorescence dye binds (indirectly) to nervous system elements. As this binding is not fully specific and there is background noise too, the presence of main nervous system portions is indicated by the intensity of the fluorescence signal. From the example it is obvious that the appearance decidedly relies on the choice/adjustment of the *colormap*: Fig. 1B shows much more unspecific background and the main components are less distinct than in Fig. 1A. At the same time there are fine nervous strings visible in Fig. 1B that are invisible in Fig. 1A. In a sense, contrast is lower in Fig. 1B as a result of *colormap* adjustment. While the advantages of Figs 1A and 1B are disputable, it is obvious that VR is more informative than the isosurface (Fig. 1C) for the present preparation. Reduced resolution is clearly discernible in the 3D PDF figure VR (Fig. 1F and 3D model), but does not severely degrade the information content and explanatory value of the figure. Figures 1D,E,G demonstrate that SR and VR can be combined in both AMIRA, and in a 3D PDF figure. Figures 1F,G exhibit further options: labels, dimensions and other annotations supported with the Adobe-supplied tool 3D REVIEWER.

Figures 1D,E are interesting in another aspect: they reveal specific problems of CLSM that usually remain unnoticed or are ignored. These are the flattening and cropping of specimens and so-called (de-)convolution. The total z-depth (thickness of the layer that can be scanned) is limited in routine CLSM to about 70–80 μm . However, most—particularly zoological—specimens are thicker than this. In routine CLSM there are two regular ways to get around this (but see e.g. Heuer & Loesel, 2009 for a sophisticated approach): scanning only part of the specimen, or compressing it, so that it fits to this range. The latter is usually accomplished by simply covering the specimen with a cover slip on the micro slide. While distance holders (e.g. pieces of a cover slip with a thickness of a little more than 100 μm) are normally used, most of the specimens become substantially physically flattened by this simple procedure. This is the case for the present example (Fig. 1): it is significantly flattened in the z direction (by a ratio of approx. 1:2) by preparation, resulting in an abnormal overall 3D arrangement. In the literature, extreme examples

with a compression ration of up to 1:10 (Altenburger & Wanninger, 2009, estimated from additional file 1) can be found. While such compression might cause some distortion in x,y arrangement, it is essentially irrelevant for the normal maximum-intensity z-projection of CLSM data. Using the data set “as is” though, for a volumetric 3D figure viewable from all angles, is of limited use.

The present example shows further methodological shortcomings: the specimen was not fully captured. The most posterior portion of the visceral loop was missed (Fig. 1D bottom-left, also visible as hole in the 3D SR figure), probably because of limited z-depth range. In contrast to z compression, such absent information is a problem for both standard and volumetric CLSM display. Another problem becomes apparent in comparing the SR (Fig. 1D) with the original (z direction) orientation (Fig. 1C): most structures appear “thicker” in the z direction than in the x and y directions (Fig. 1D). This is due to instrumental convolution (see McNally et al., 1999 for principles), which is a blurring of images in all directions. It is most prominent in the z direction. Put another way, the imaging resolution is normally poorer in the z direction than in the x and y directions. Convolution is more of a problem in normal (“wide-field”) microscopy, but is also present in CLSM. There are deconvolution software tools to reduce this blurring (e.g. in AMIRA) which were not applied here. Again, as for specimen flattening, convolution is unimportant for standard 2D projections of CLSM data along the z direction.

Summing up, there are important limitations for utilizing CLSM data of biological specimens for 3D visualizations other than standard 2D (MIP) projections, like, for example, changing the viewing angle. We critically mention that information on specimen mounting (and thus degree of flattening) and physical sample dimensions (and thus z-depth) are generally withheld (eg. Wollesen et al., 2007; Gruhl, 2009) in studies of animal morphology. While it is normal to provide a scale bar for x and y dimensions, information on the z-dimension of an image is rarely given, although there are positive exceptions (e.g. Maslakova & von Döhren, 2009). Presenting this methodological information should be obligatory, as it is often crucial for the correct interpretation of data.

Example 2 (Fig. 2): This example can be compared to the same specimen treated in Ruthensteiner & Heß (2008) with manual segmentation SR. VRs (Fig. 2A,B,D) provide a good overall impression of the body surface as well as fine details. This

can be seen by comparison to both the isosurface (Figs 2C,F) and the manual segmentation surface (Ruthensteiner & Heß, 2008, Fig. 3A). At the downsampled resolution of the 3D PDF figure (Fig. 2D and 3D model) there is some loss of finer details, such as ciliation or operculum, but this model conveys useful information on internal components. At certain rotation angles of the 3D model, inconsistent layers become visible. These result from artefacts in the original data such as different staining intensities of sections, leading to different offsets or levels in the acquired data sections. Nevertheless, as is evident from the present example, VR for visualization of serial sections can be a valuable complement to SR.

The AMIRA isosurface (Fig. 2C) compares well to the manual segmentation surface (Ruthensteiner & Heß, 2008, Fig. 3A), while reduction of tessellation to yield a conveniently small size for the PDF 3D isosurface (Fig. 2F) causes some degradation. Since it is desirable to keep file size small, for the present example the manual segmentation surface is preferred over the isosurface for the 3D PDF figure. Despite the increasing prevalence of tomographic methods, serial section analysis will continue to be a principal morphological method in the future, because it yields a wide range of detailed information. The chief attention, however, is not going to be the construction of interactive, 3D VRs. The example presented herein rather is intended to demonstrate that VR can be useful and informative as a supplementary visualization mode. Effort involved in the graphical procedure (separate image acquisition, image cleaning, image alignment) is greater than for other methods. Furthermore, it is particularly difficult to discretely show specific structures in VR. Nevertheless, if a section has already been prepared for SR, then VR can be executed with minor additional effort, and if implemented in a 3D PDF figure it provides an additional, unique way to share data visualisations more openly with the community.

Example 3 (Fig. 3): a small dry shell plate of a chiton is an ideal object for microCT. Fine structures, such as the inner tubing system, are best shown by VR (Figs 3C,D,F,G), whereas the whole structure, the shell surface, is best shown by SR. While the 3D PDF VR (Figs 3F,G) still contains the central information—the main arrangement is recognizable—some blurring caused by reduction of resolution is obvious. Additionally in this case, the intrinsically diffuse nature of the VR gives rise to an *apparent* volume increase, ie. the slight projection of the volume beyond

the shell surface (Fig. 3E). Such effects may be desirable, depending on the intent of the scientist.

There are no obstacles such as dimensional distortion or convolution effects as present in CLSM—microCT-data are ideally suited to direct 3D visualisation. This is very convenient as microCT is becoming increasingly popular in biomedical science. Because of its capability to image soft tissues, as demonstrated by Metscher (2009), it is predestined to become standard for a wide range of topics.

General—relative merits of SR and VR

There is a long history, mostly dealing with examples from applied medical research, of weighing up merits and obstacles of SR and VR (e.g. Udupa et al. 1991; Calhoun et al., 1999). Conclusions vary (Kuszyk et al., 1999 vs. Rodt et al., 2006) and clearly the best choice of visualization mode very much depends on the individual problem. This also applies to the examples presented herein. In general though, we recommend SR for interpreting or emphasizing physical or logical *components* of a 3D data set (e.g. Ruthensteiner & Heß, 2008) while VR is preferred for *global* overviews that, with careful choice of colour and opacity mapping, can simultaneously reveal internal structure. In any case, VR is complementary to SR and we contend that it is clearly valuable to have the option to embed interactive 3D VR figures as well as interactive 3D SR figures in research articles.

General—embedded 3D PDF figures and VR

As discussed above SR is inadequate for many biomedical purposes; it is inadequate for naturally displaying many types of biological data. Consequently, the inclusion of interactive, 3D VR figures in PDFs, as demonstrated by Barnes & Fluke (2008), is highly desirable for biomedical purposes as well. The technical implementation of VR, however, is more difficult than that of SR, where models—created by various software applications—of a wide range of formats can be incorporated in the PDF. One way to put this into practice is the S2PLOT library (Barnes et al., 2006; Barnes & Fluke, 2008), from which scenes can be exported to VRML format suitable for incorporation in PDF documents. This technique has been utilized for the present paper.

The creation of an interactive, 3D VR figure in PDF entails the inclusion of three stacks of PNG images together with a VRML file and a JavaScript 3D program code as described earlier. It is useful to briefly consider the resultant file size. For the *Acroloxus lacustris* rendering (Fig. 1), the downsampled volume dimensions are 186 x 168 x 139 voxels. A single, *uncompressed* image stack requires 186 x 168 x 139 voxels x 4 bytes-per-voxel = ~16M bytes (MB) of storage space; the PDF must store three image stacks (one per axis), a total of 48 MB. Fortunately, we are able to store the image stacks as losslessly compressed PNG files. For *A. lacustris* the PNG compression is a factor of around 25 times, yielding an acceptable PDF figure size of approximately 1.8 MB. The level of compression achieved in PNG files is image dependent though, and balancing of the level of downsampling prior to figure creation, against achieving the required resolution in the output figure, will usually be a routine choice to make in generating the 3D figure.

Direct comparison of the file sizes of SR and VR visualisations as 3D PDF figures is problematic, even if they are derived from the same data. The file size of the SR PDF figure will depend principally on the degree of tessellation and the complexity of the surface (and to a lesser extent on properties such as data dimensions and signal-to-noise-level). Conversely, the file size of the VR PDF figure will depend mostly on the data dimensions, the noise level (noise does not compress well), and the fraction of the volume which holds non-zero data values. The choice of properties in creating the figures is a matter of subjective intention of what is to be shown, and so a numerical comparison is of limited value. Nevertheless, we can at least report the file size for the VR data of our models (in total: size of the *Poster Image*—the image visible when 3D content is disabled—plus all 3D components). These are approximately 1.8 MB for example 1, 3.6 MB for example 2 and 0.6 MB for example 3. The VR data plus all additional data of the paper adds up to less than 11 MB. This is large for a scientific PDF publication but not unusual or prohibitive. Transfer of files this size over the Internet, via HTTP, FTP or e-mail protocols is routine, and permitted in most scientific institutions.

Conclusion

We have demonstrated that volume rendering of biological objects can be used for interactive 3D figures in PDF files, using data collected from a wide range of

morphological techniques. By example we have shown that SR and VR are complementary techniques and there is a place for both in the scientific literature. The impact on file size of including interactive, 3D VR figures in PDF publications is noticeable—typically 2 or 3 MB per VR figure—but can be managed by careful balancing of the input data size and the didactic value of the final figure. We advocate that 3D models using VR should become standard in PDF versions of biomedical publications, as devices for sharing instructive data visualisations with the scientific community.

Acknowledgments

We are indebted to those who provided help and access to sophisticated gear: Charles David and his working group (LMU, Munich, Department Biologie II)—CLSM; Karl-Heinz Kunzelmann and his working group (LMU, Munich, Poliklinik für Zahnerhaltung und Parodontologie)—microCT. Thanks to Dirk Eheberg for help with immunofluorescence preparations. Special thanks go to Gerhard Haszprunar (Bavarian State Collection of Zoology) for supervising the diploma thesis of NB within the scope of which the CLSM data have originated. We thank the GeoBio-Center of the LMU Munich for providing a license of AMIRA.

References

- Altenburger, A., Wanninger, A., 2009. Comparative larval myogenesis and adult myoanatomy of the rhynchonelliform (articulate) brachiopods *Argyrotheca cordata*, *A. cistellula*, and *Terebratalia transversa*. *Front. Zool.* 6, 3 doi:10.1186/1742-9994-6-3.
- Baeumler, N., Haszprunar, G., Ruthensteiner, B., 2008. 3D interactive microanatomy of *Omalogyra atomus* (Philippi, 1841) (Gastropoda, Heterobranchia, Omalogyridae). *Zoosymposia.* 1, 101–118.
- Barnes, D.G., Fluke, C.J., Bourke, P.D., Parry, O.T., 2006. An advanced, three-dimensional plotting library for astronomy. *Publications of the Astronomical Society of Australia.* 23, 82–93.

- Barnes, D.G., Fluke, C.J., 2008. Incorporating interactive 3-dimensional graphics in astronomy research papers. *New Astronomy*, 13, 599–605.
- Calhoun, P.S., Kuszyk, B.S., Heath, D.G., Carley, J.C., Fishman, E.K. 1999. Three-dimensional Volume Rendering of Spiral CT Data: Theory and Method. *Radiographics*. 19, 745–764.
- Claxton, N.S., Fellers, T.J., Davidson, M.W., 2008. Laser scanning confocal microscopy. 37 pp. <http://www.olympusconfocal.com/theory/LSCMIntro.pdf>.
- Dreben, R., Carpenter, L., Hanrahan, P., 1988. Volume rendering. *Comp. Graph*. 22, 65–74.
- Gruhl, A. 2009. Serotonergic and FMRFamideergic nervous systems in gymnolaemate bryozoan larvae. *Zoomorphology*. 128, 135–156.
- Evans, C.C.E., Dickinson, A.J.G., Croll, R.P., 2009. Major muscle systems in the larval caenogastropod, *Ilyanassa obsoleta*, display different patterns of development. *J. Morph.* 270, 1219–1231.
- Fernandez, C.Z., Vendrasco, M.J., Runnegar, B., 2007. Aesthete canal morphology in twelve species of chiton (Polyplacophora). *Veliger*. 49, 51–69.
- Fishman, E.K. Ney, D.R., Heath, D.G., Corl, F.M., Horton, K.M., Johnson, P.T. 2006. Volume Rendering versus Maximum Intensity Projection in CT Angiography: What Works Best, When, and Why. *Radiographics*. 26, 905–922.
- Fuchs, H., Kedem, Z.M., Uselton, S.P. 1977. Optimal Surface Reconstruction for Planar Contours. *Commun. ACM*. 20, 693–702.
- Golding, R.E., Ponder, W.F., Byrne, M., 2009. Three-dimensional reconstruction of the odontophoral cartilages of Caenogastropoda (Mollusca: Gastropoda) using micro-CT: morphology and phylogenetic significance. *J. Morph.* 270, 558–587.
- Heuer, C. M. & Loesel, R. (2009) Three-dimensional reconstruction of mushroom body neuropils in the polychaete species *Nereis diversicolor* and *Harmothoe areolata*. (Phyllodocida, Annelida). *Zoomorphology*, 128, 219–226.
- Kaufman, A., Mueller, K. 2005. Overview of Volume Rendering, in: Hansen, C., Johnson, C.R. (Eds.), *The Visualization Handbook*. Academic Press, London, pp. 127–174.
- Kumar, P., Ziegler, A., Ziegler, J., Uchanska-Ziegler, B., Ziegler, A., 2008. Grasping molecular structures through publication-integrated 3D models. *Trends Biochem. Sci.* 33, 408–412.

- Kuszyk, B.S., Heath, D.G., Bliss, D.F., Fishman, E.K. 1996. Skeletal 3-D CT: advantages of volume rendering over surface rendering. *Skeletal Radiol.* 25, 207–214.
- Maslakova, S.A., von Döhren, J., 2009. Larval Development with transitory epidermis in *Paranemertes peregrina* and other hoplonemertans. *Biol. Bull.* 216, 273–292.
- McNally, J.G., Karpova, T., Cooper, J., Conchello, J.A., 1999. Three-dimensional imaging by deconvolution microscopy. *Methods.* 19, 373–385.
- Metscher, B.D., 2009. MicroCT for comparative morphology: simple staining methods allow high-contrast 3D imaging of diverse non-mineralized animal Tissues. *BMC Physiol.* 9, 11, doi:10.1186/1472-6793-9-11.
- Murienne, J., Ziegler, A., Ruthensteiner, B., 2008. A 3D revolution in communicating science. *Nature.* 453, 450.
- Neues, F., Epple, M., 2008. X-ray micro computer tomography for the study of biomineralized endo- and exoskeletons of animals. *Chem. Rev.* 108, 4734–4741.
- Paddock, S.W., 1999. 1 An introduction to confocal imaging. in: Paddock, S.W. (Ed.), *Confocal Microscopy Methods and Protocols.* Volume 122 in the series *Methods in Molecular Biology.* Walker, J.M. (series Ed.). Humana Press, Totowa, NJ, pp. 1–34.
- Pietsch, T.W., Johnson, J.W., Arnold, R.J. 2009. A New Genus and Species of the Shallow-Water Anglerfish Family Tetrabrachiidae (Teleostei: Lophiiformes: Antennarioidei) from Australia and Indonesia. *Copeia.* 3, 483–493.
- Ribi, W., Senden, T.J., Sakellariou, A., Limaye, A., Zhang, S., 2008. Imaging honey bee brain anatomy with micro-X-ray-computed tomography. *J. Neurosci. Methods.* 171, 93–97.
- Ritman, E.L., 2004. Micro-computed tomography—current status and developments. *Annu. Rev. Biomed. Eng.* 6, 185–208.
- Rodt, T., Bartling, S.O., Zajaczek, J.E., Vafa, M.A., Kapapa, T., Majdani, O., Krauss, J.K., Zumkeller, M., Matthies, H., Becker, H., Kaminsky, J. 2006. Evaluation of surface and volume rendering in 3D-CT of facial fractures. *Dentomaxillofac. Radiol.* 35, 227–231.
- Ruffins, S.W., Martin, M., Keough, L., Truong, S., Fraser, S.E., Jacobs, R.E., Lansford, R. 2007. Digital Three-Dimensional Atlas of Quail Development Using High-Resolution MRI. *TSW Dev. Embryol.* 2, 47–59.
- Ruthensteiner, B., Heß, M., 2008. Embedding 3D models of biological specimens in PDF publications. *Microsc. Res. Tech.* 71, 778–786.

- Ruthensteiner, B., 2008. Soft Part 3D visualization by serial sectioning and computer reconstruction. *Zoosymposia*. 1, 63–100.
- Schwaha, T., Gith, R., Walzl, M.G., 2008. The nutritive region in the ovaries of astigmatic mites (Acari: Acaridida). *Soil Org.* 80, 249–259.
- Storck, S.R., 2009. *MicroComputed Tomography: Methodology and Applications*. CRC Press, Taylor & Francis Group, Boca Raton, Fl.
- Truong, V.K., Rundell, S., Lapovok, R., Estrin, Y., Wang, J.Y., Berndt, C.C., Barnes, D.G., Fluke, C.J., Crawford, R.J., Ivanova, E.P., 2009. Effect of ultrafine-grained titanium surfaces on adhesion of bacteria. *App. Microbial and Cell Phys.* 83, 925–937.
- Tyzack, J.K., 2008. Editorial—dragging (and zooming and rotating) publication of 3D molecular structures into the 21st century. *Trends Biochem. Sci.* 33, 405–407.
- Udupa J.K., Hung, H.M., Chuang, K.S. 1991. Surface and volume rendering in three-dimensional imaging: a comparison. *J. Digit. Imaging.* 4, 159–168.
- Webb, A.G., 2003. X-Ray imaging and computed tomography. In: *Introduction to Biomedical Imaging*. Wiley-IEEE Press, Hoboken, NJ, pp. 1–56.
- Wirkner, C.S., Richter, S., 2008. Morphology of the haemolymph vascular system in Tanaidacea and Cumacea: —Implications for the relationships of “core group” Peracarida (Malacostraca; Crustacea). *Arthropod Struct. Dev.* 37, 141–154.
- Wollesen, T., Wanninger, A., Klusmann-Kolb, A., 2007. Neurogenesis of cephalic sensory organs of *Aplysia californica*. *Cell Tissue Res.* 330, 361–379.
- Ziegler, A., Faber, C., Mueller, S., Bartolomaeus, T., 2008. Systematic comparison and reconstruction of sea urchin (Echinoidea) internal anatomy: a novel approach using magnetic resonance imaging. *BMC Biol.* 6, 33, doi:10.1186/1741-7007-6-33.

Table 1. 3D volume sizes and physical dimensions (voxel size in μm x, y, z) of model data sets.

	Example 1	Example 2	Example 3
Raw data size (voxels)	1024, 1024, 139	1600, 1200, 219	940, 433, 384
Raw data physical voxel dimensions	0.39, 0.39, 0.49	0.095, 0.095, 0.75	6, 6, 6
Cropped size (AMIRA image stack)	741, 672, 139	1820, 1155, 219	612, 286, 335
Downsampled size (PDF 3D model)	186, 168, 139	182, 116, 219	307, 143, 167
Downsampling ratio (PDF 3D model)	0.25:0.25:1	0.1:0.1:1	0.5:0.5:0.5

Figure captions:

Fig. 1. *Acroloxus lacustris*. 13 days old juvenile, early creeping stage. Part of the specimen with serotonergic central nervous system at different visualization modes and views from CLSM data. A–C,F. from the left; D. from ventral; E. from ventral left; G. from posterior-left. A–E. AMIRA visualizations. A,B. *Voltex* VR with two different *colormap* adjustments. C. Isosurface (downsampled 4/4/1, threshold ca. 75, triangles reduced to ca. 24,000). D,E. VR (settings like in B) plus isosurface. F,G. PDF 3D visualizations (also represented as views in the 3D model) generated with S2PLOT. F. VR with 3D labels for nervous system components. 3D labels (and 3D dimensions in G.) are shown only to draw attention to particular aspects of the 3D model. For structural labelling see A. G. Volume rendering (same as F.) plus isosurface with 3D bounding box and 3D dimensions in μm . Abbreviations: a— anterior, cc—cerebral commissure, cg—cerebral ganglia, cpc—cerebropedal connective, d—dorsal, lpn—left pallial nerve, npf—nerve plexus of foot, p— posterior, pg—pedal ganglia, rpn—right pallial nerve, v—ventral, v2—visceral nerve 2, vl—visceral loop. Scale bar (in C—same for A–F) = 50 μm .

When viewing with Adobe Reader (recent version recommended), click the image to activate the 3D mode. A variety of tools allows interactive manipulations ranging from free rotating (Rotate (default) or Spin option activated on 3D bar, and drag model with mouse), zooming in and out (Zoom option activated) or moving the model (Pan option activated) to change the appearance by viewing only selected portions (use Model Tree - check or uncheck objects) or changing surface visualization (Render Mode, Lightning). In addition to the default view, custom views (with self-explanatory names) are provided and available by clicking on them in the middle of the Model Tree interface part. In these views, information like rotation angle, magnification, portion selection or transparency are saved. Lighting or Render Mode initially appear as set by default, but can be adjusted independently of the views. See also Tyzack (2008) for handling of 3D models. Note that some choices of lighting or rendering mode may impair visibility of some components of the model (especially VR).

Fig. 2. *Patella caerulea*. Metamorphic competent larva. Total specimen at different visualization modes and views from serial section data. A–D. from the left; F. from

the right; A–C. AMIRA visualizations. A, B. *Vol/tex* VR visualizations with different *colormaps*. Downsampled 2/2/1. A. Alpha scale: 0.5; B. Alpha scale: 1. C. Isosurface. Downsampled 2/2/1, threshold 30, extracted surface smoothed with high (40) iteration value. E. Slice (nr. 98) taken from the AMIRA 2D TIF-format file stack, left—unaltered from original stack, right—inverted, enhanced and cleaned for VR. D,F. PDF 3D visualizations (also represented as views in the 3D model). D. VR generated with S2PLOT with 3D labels for morphological components. For structural labelling see B,E. F. Isosurface generated with AMIRA (downsampled 6/6/1, triangles of extracted surface reduced to about 10,000). Abbreviations: a—anterior, cg—cerebral ganglia, d—dorsal, o—operculum, mc—mantle cavity, me—mantle edge, p—posterior, pc—prototroch cilia, ps—pedal sole, pt—prototroch, rm—retractor muscles, te—tentacle, v—ventral. Scale bar (B—same for A–C,F) = 50 μm . See Fig. 1 for viewing 3D mode with Adobe Reader.

Fig. 3. *Lepidochitona corrugata*. Shell plate at different visualization modes and views from microCT data. A, B. Identical crop of section, before (A) and after (B) image treatment for VR. C,D. AMIRA *vol/tex* VR with original (cropped) resolution. Area framed in C is enlarged in D. E–G. PDF 3D visualizations (also represented as views in the 3D model). Shell plate surface (E., created with AMIRA) plus VR (E–G.) of internal tubing system generated with S2PLOT. Magnification and rotation angle are identical to C (equals E,F.) and D (equals G.). a—anterior, cl—centreline, es—external surface (dorsal towards the exterior), it—internal tubing system, is—internal surface (ventral towards the soft body), p—posterior. Scale bar (in C.) = 500 μm . See Fig. 1 for viewing 3D mode with Adobe Reader.

Appendix A—VOLREN program code (demonstrated at *Acroloxus lacustris*—example 1)

The VOLREN program code is presented below (Listing A). It can be compiled on a modern Apple Mac OS X or GNU Linux system, with an up-to-date installation of S2PLOT (available from <http://astronomy.swin.edu.au/s2plot>) by entering the command:

```
cbuild.csh volren.c
```

The resultant executable program expects to read a set of 139 greyscale TGA files named according to the format “`slice-nnn.tga`” and stored in the folder “`volume`”.

The preface section of the program code includes the necessary header files, and defines the following global variables of note: `volume` to store the data read in from the image stacks, and `vid1`, `vid2` and `vid3` to store references to the three unique volume renderings obtained by slicing the volume in the three axial directions.

The `initVolume` function is called once by the main program to read in the stack of TGA image files and reassemble the volume as a contiguous set of voxels in the memory referenced by the global variable `volume`.

The `drawVolume` function is called once every redraw cycle by S2PLOT, and uses the (hidden) S2PLOT function `ds2dvrXXX` to forcibly draw all three volume renderings to the screen. That is, the set of slices along the X-axis are drawn to the screen in back-to-front order, followed by the set of volume slices along the Y-axis, and finally the Z-axis. The calls to `ds2dvrXXX` are spaced by calls to `pushVRMLname`, which is the S2PLOT function that allows us to label parts of the drawing so that they will ultimately appear as separate sections of the model tree when a VRML file is produced by S2PLOT. For normal interactive volume rendering in a standalone S2PLOT program where VRML (and 3D PDF) is not the purpose, the code shown in `drawVolume` below would be replaced by a singular call to the standard `ds2dvr` function.

The `alphafn` function is defined to provide a non-linear mapping from data value to opacity for the actual volume rendering. In this particular example, opacity for a given voxel ranges from 0.002 to 0.7 and is proportional to the cube root of the data value. Accordingly, high data values are less transparent (more opaque) than low

data values. This function is registered in the call(s) to `ns2cvra` in the main program.

In the main program code of VOLREN, the main view coordinate system and transformation matrix are configured. In this particular case, as our intention is to include accurate dimensions, labels and a bounding box, we have been very specific in our setup of the coordinate system. Ordinarily for interactive volume rendering the transformation matrix (`tr`), the viewport (set with the function call `s2svp`) and the world coordinate system (`s2swin`) can be established in significantly less code. As a final step before instructing S2PLOT to begin drawing the volume rendering(s) and to allow interaction by the user, we install a standard S2PLOT colourmap called “mgreen” (the `s2icm` function). The call to `ns2sevas` is important: it sets a relative scaling of opacities for views along the three axes. As we are creating a volume rendering for VRML (and 3D PDF), we are actually making three volume renderings, only one of which will be shown at any particular time in the PDF view. Because the original data volume is not cubic (ie. it has unequal side lengths) we need to globally increase opacity when a “shorter” stack of slices is being viewed, to create the same apparent volume density. This is the function of the `ns2sevas` function and the choice of scaling values should be guided by the original data dimensions as well as the explicit scaling applied in the `alphafn` function.

With the colourmap installed and opacity scaling set, the main program code now simply calls the `ns2cvra` function three times to generate the axial volume renderings, registers the `drawvolume` function, and hands over control to S2PLOT by calling the `s2show` function.

Listing A. VOLREN program code.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <ctype.h>
#include "s2plot.h"
#include "bitmaplib.h"
#include "opengl1lib.h"
/* "hidden" S2PLOT function for volume rendering to VRML */
void ds2dvrXXX(int, int, int);

/* Global variables */
float ***volume; /* the data */
int vid1, vid2, vid3; /* ids for volume slice sets along x,y,z axes */
float tr[12]; /* transformation matrix */

// input file description
#define NAMEFORM "volume/slice-%03d.tga"
```

```

#define FIRSTFRAME 0
#define LASTFRAME 138
#define STRIDE 1

// data dimensions in voxels
#define ORIG_NW 742
#define ORIG_NH 673
#define ORIG_ND 139

// voxel dimensions in microns
#define ORIG_VW 0.390625
#define ORIG_VH 0.390625
#define ORIG_VD 0.488440

// volume rendering data range
#define DMIN 30.0
#define DMAX 350.0

// volume rendering alpha (transparency) range
#define AMIN 0.002
#define AMAX 0.7

/* allocate memory and load the data cube from a set of slices */
float ***initVolume(int *nx, int *ny, int *nz) {
    char fname[255];
    BITMAP4 *readbits;
    float ***retvol = 0;
    int i, j, k, w, h;
    *nz = (LASTFRAME - FIRSTFRAME + 1) / STRIDE;

    for (i = 0; i < (LASTFRAME-FIRSTFRAME+1); i+=STRIDE) {
        sprintf(fname, NAMEFORM, FIRSTFRAME + i);
        readbits = ReadTGATexture(fname, &w, &h);
        if (!readbits) {
            fprintf(stderr, "Failed to read image %s\n", fname);
            exit(-1);
        }
        if (i == 0) { /* allocate memory when first frame read */
            *nx = w;
            *ny = h;
            retvol = (float ***)malloc(*nx * sizeof(float **));
            if (retvol == NULL) {
                fprintf(stderr, "Failed to allocate %ld bytes\n",
                    (long)(*nx*sizeof(float **)));
                exit(-1);
            }
            for (j = 0; j < *nx; j++) {
                retvol[j] = (float **)malloc(*ny * sizeof(float *));
                for (k = 0; k < *ny; k++) {
                    retvol[j][k] = (float *)malloc((*nz) * sizeof(float));
                }
            }
        }
        for (j = 0; j < *nx; j++) { /* copy this slice into volume */
            for (k = 0; k < *ny; k++) {
                retvol[j][*ny-1-k][i/STRIDE] = readbits[j + k * w].r +
                    readbits[j + k * w].g + readbits[j + k * w].b;
            }
        }
    }
    return retvol;
}

/* called every draw cycle to draw the three volume renderings */
void drawvolume(double *t, int *kc) {
    ss2srm(SHADE_FLAT); /* set the shading style to flat */
    pushVRMLname("VRSET1");
    ds2dvrXXX(vid1, 1, 1); /* draw the x-aligned set of slices */
    pushVRMLname("VRSET2");
    ds2dvrXXX(vid2, 1, 2); /* draw the y-aligned set of slices */
    pushVRMLname("VRSET3");
    ds2dvrXXX(vid3, 1, 3); /* draw the z-aligned set of slices */
    pushVRMLname("ANON");
}

/* function to provide alpha (opacity) as a function of data value */

```

```

float alphafn(float *dval) {
    float valsca = (*dval - DMIN) / (DMAX - DMIN);
    valsca = valsca < 0.0 ? 0.0 : valsca > 1.0 ? 1.0 : valsca;
    return AMIN + (AMAX - AMIN) * powf(valsca, 0.33);
}

/* main program function */
int main(int argc, char *argv[]) {
    int i; /* a useful loop variable */
    int nx, ny, nz; /* dimensions of data cube */
    float x1 = 0, x2 = +1; /* define the coordinates we will use */
    float y1 = 0, y2 = +1;
    float z1 = 0, z2 = +1;

    volume = initVolume(&nx, &ny, &nz); /* read the data */

    for (i = 0; i < 12; i++) { tr[i] = 0.0; } /* setup transform matrix */
    tr[ 0] = x1; tr[ 1] = (x2-x1)/(float)(nx-1.0);
    tr[ 4] = y1; tr[ 6] = (y2-y1)/(float)(ny-1.0);
    tr[ 8] = z1; tr[11] = (z2-z1)/(float)(nz-1.0);

    s2opend("/?",argc,argv); /* open the display */
    s2svp(0.0, (float)(ORIG_NW) * ORIG_VW, /* set the 3-d viewport */
         0.0, (float)(ORIG_NH) * ORIG_VH,
         0.0, (float)(ORIG_ND) * ORIG_VD);
    s2swin(x1, x2, y1, y2, z1, z2); /* set the 3-d coordinates */

    pushVRMLname("BOUNDING_BOX");
    s2box("BCDE",0,0,"BCDE",0,0,"BCDE",0,0); /* draw the bounding box */
    pushVRMLname("ANON");

    s2scir(1000,2000); /* install colourmap */
    s2icm("mgreen",500,2000);
    ns2sevas(1.0, 1.0, 0.3); /* set opacity scaling */

    /* create three volume renderings (one for each principal axial view) */
    vid1 = ns2cvra(volume, nx, ny, nz, 0, nx-1, 0, ny-1, 0, nz-1,
                  tr, 't', DMIN, DMAX, alphafn);
    vid2 = ns2cvra(volume, nx, ny, nz, 0, nx-1, 0, ny-1, 0, nz-1,
                  tr, 't', DMIN, DMAX, alphafn);
    vid3 = ns2cvra(volume, nx, ny, nz, 0, nx-1, 0, ny-1, 0, nz-1,
                  tr, 't', DMIN, DMAX, alphafn);

    cs2scb(drawvolume); /* register the function that draws the volume */
    s2show(1); /* display the visualisation */

    return 1;
}

```

Appendix B—3D JavaScript extract

An extract of the 3D JavaScript which should be included in S2PLOT-created 3D VR figures in PDF, is presented below (Listing B). Its primary function is to select one of the three sets of slices that should be drawn to provide the best view of the data as the viewing (camera) angle changes. The full script (available from <http://astronomy.swin.edu.au/s2plot>) includes a basic implementation of the standard S2PLOT keyboard and mouse interface for camera control: these capabilities are not discussed here.

The VRML model exported by the S2PLOT code (eg. the VOLREN program given in Appendix A) includes 3 volume renderings stored in separate “branches” of the model tree. The “leaves” of the tree are individual graphical elements, ie. single

textured facets that make up the volume renderings. For volume rendering, the branches are identified by the labels VRSET1, VRSET2 and VRSET3. The tree structure is maintained in the 3D PDF figure, and in the first section of the script extract, commencing with “`// compile lists of VRSET* meshes`”, the model tree is examined and lists of the leaves making up each of the branches VRSET1, VRSET2 and VRSET3 are compiled. This is done once, when the 3D mode is enabled, normally by the user clicking on the figure’s poster image.

Three functions are then defined in the script. The first function—`setVRSETvis`—toggles the visibility of all the leaves in a specific list, according to the arguments given to the function. The second function—`allSetVis`—is a utility function which simply calls `setVRSETvis` three times to make one set of leaves visible and the other two invisible. The third function—`pickFrameSet`—is the function which examines the current viewing angle and selects which branch of the tree (VRSET1, VRSET2, VRSET3) is the best to draw. The choice is made by selecting the set of slices that are “layered” along the Cartesian axis (x, y or z) that is nearest to parallel to the line from the camera to the centre of the rendering. For data sets that are approximately cubic (in voxel dimensions), this strategy effectively selects the densest sampling of slices along the view direction, and so yields the most smoothest rendering.

At the end of the script extract, two “handlers” are installed that instruct the PDF viewer to call the `pickFrameSet` function whenever the mouse is dragged with the left button down, or the scene is rendered.

Finally, the script explicitly calls `pickFrameSet` once, so that when the figure is first viewed in 3D mode, the rendering is updated for the initial camera position and viewing direction.

Listing B. S2PLOT.JS script extract.

```
// global variables to keep lists of meshes
var VRSET1objsArray = new Array();
var VRSET1objsCount = 0;
var VRSET2objsArray = new Array();
var VRSET2objsCount = 0;
var VRSET3objsArray = new Array();
var VRSET3objsCount = 0;

// compile lists of VRSET* meshes
for (i=0; i<scene.meshes.count; i++) {
  currMesh = scene.meshes.getByIndex(i);
  currParentName = currMesh.parent.name.toString();
  currPPName = currMesh.parent.parent.name.toString();
  currPPPName = currMesh.parent.parent.parent.name.toString();
  currPPPPName = currMesh.parent.parent.parent.parent.name.toString();
  framestr = "";
  if (( currParentName.indexOf("VRSET1.") > -1) ||
```

```

        ( currPPname.indexOf("VRSET1.") > -1 ) ||
        ( currPPPname.indexOf("VRSET1.") > -1) ) {
    VRSET1objsArray[VRSET1objsCount] = currMesh;
    VRSET1objsCount++;
} else if ( ( currParentName.indexOf("VRSET2.") > -1) ||
( currPPname.indexOf("VRSET2.") > -1) ||
( currPPPname.indexOf("VRSET2.") > -1) ) {
    VRSET2objsArray[VRSET2objsCount] = currMesh;
    VRSET2objsCount++;
} else if ( ( currParentName.indexOf("VRSET3.") > -1) ||
( currPPname.indexOf("VRSET3.") > -1) ||
( currPPPname.indexOf("VRSET3.") > -1) ) {
    VRSET3objsArray[VRSET3objsCount] = currMesh;
    VRSET3objsCount++;
}
}

// volume rendering handling code function
setVRSETvis(which, visibility, update) {
    if (which == 0) {
        for (j=0; j<VRSET1objsArray.length; j++) {
            VRSET1objsArray[j].visible = visibility;
        }
    } else if (which == 1) {
        for (j=0; j<VRSET2objsArray.length; j++) {
            VRSET2objsArray[j].visible = visibility;
        }
    } else if (which == 2) {
        for (j=0; j<VRSET3objsArray.length; j++) {
            VRSET3objsArray[j].visible = visibility;
        }
    }
    if (update) {
        scene.update();
        runtime.refresh();
    }
}

function allSetVis(which) {
    setVRSETvis(which, true, 0);
    setVRSETvis((which + 1) % 3, false, 0);
    setVRSETvis((which + 2) % 3, false, 1);
}

function pickFrameSet() {
    var camera = scene.cameras.getByIndex(0);
    var camdir = camera.position.subtract(camera.targetPosition);

    // default
    var whichframeset = 0;
    if ((Math.abs(camdir.y) >= Math.abs(camdir.x)) &&
        (Math.abs(camdir.y) >= Math.abs(camdir.z))) {
        whichframeset = 1;
    } else if ((Math.abs(camdir.z) >= Math.abs(camdir.x)) &&
        (Math.abs(camdir.z) >= Math.abs(camdir.y))) {
        whichframeset = 2;
    }
    allSetVis(whichframeset);
}

mreh = new MouseEventHandler();
mreh.onMouseDown = false;
mreh.onMouseMove = true;
mreh.reportAllTargets = false;
mreh.onEvent = function(event) {
    if (event.leftButtonDown) {
        pickFrameSet();
    }
}
runtime.addEventHandler(mreh);

ceh = new RenderEventHandler();
ceh.onEvent = function(event) {
    pickFrameSet();
}
runtime.addEventHandler(ceh);
pickFrameSet();

```

e-component

[Click here to download e-component: Ruthenst-etal_BVR_3D-models.pdf](#)

Figure [Click here to download high resolution image](#)

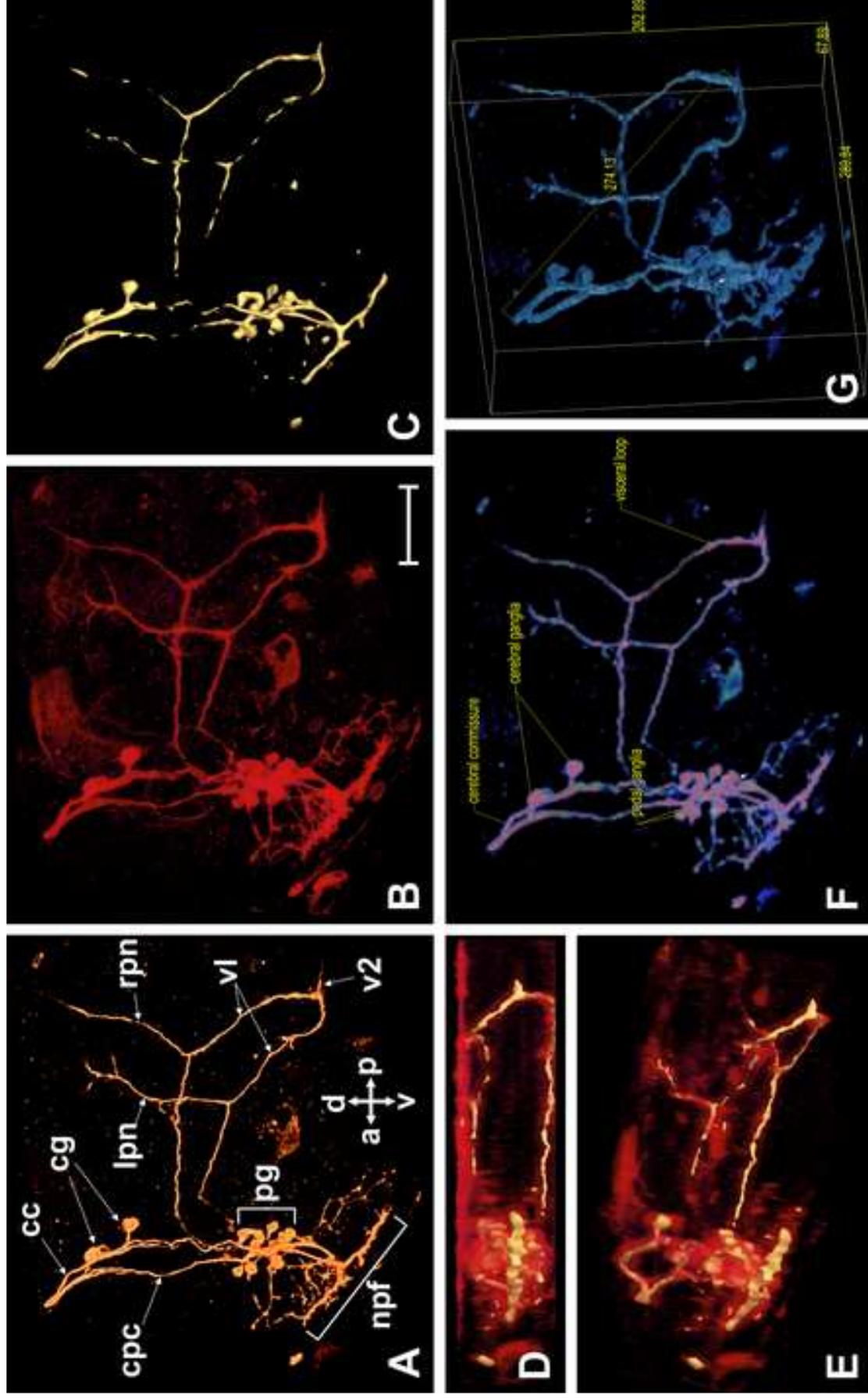


Figure
[Click here to download high resolution image](#)

