Supplementary 3D-pdf protocol
Version 10-09

Publishing three-dimensional structural information through Adobe's Portable Document Format (pdf); A Protocol for Biologists

METHOD

The programs we use in this protocol are Amira 5.2, Adobe Acrobat Pro Extended 9.3, Adobe Reader 9.3 and Adobe Illustrator CS4. From here on we will refer to these programs as Amira, Acrobat, Adobe Reader and Illustrator, respectively. Furthermore, we will refer to the 3D object as **object** (e.g. a 3D reconstruction of a heart) and the parts this object is composed of as **structure(s)** (e.g. myocardium and lumen), both in Amira and Acrobat. The protocol assumes that the user is already familiar with creating 3D reconstructions. Although this protocol starts with a 3D reconstruction created in Amira, it can be adapted for other 3D visualisation programs.

The protocol consists of two parts. Firstly, a *basic protocol* describes how to embed a 3D surface reconstruction from Amira into a pdf. It describes how to configure Amira and Acrobat and how to properly export the 3D surfaces from Amira into a pdf. This pdf can then be saved as a basic 3D-pdf (Fig. S2) and the embedded 3D surface (we will refer to it as object) can be handled interactively using the Adobe Reader user interface (Fig. S1).

The second part of the protocol describes *advanced steps* requiring a little more computer and scripting expertise. This part describes five (independent) additions; 1) creating *views and annotations* of the object, 2) applying *layout* to the design, 3) adding *buttons*, *hyperlinks* and 4) *scripts* for interaction and functionality; and 5) preparing and displaying *cross sections*. A pdf containing these additional steps will be referred to as an advanced pdf (Fig. S3).

Finally, we would like to note that possible updates to the protocols and scripts described herein will be made available at the download page of http://3d.hfrc.nl.

Please refer to the main publication when a protocol or script described herein has been applied.

BASIC PROTOCOL

Configure Acrobat

- Start Acrobat and open the *3D Capture* window by selecting *File* > *Create PDF* > *From 3D Capture* from the File menu.
- Start Amira and load and display a simple surface. Select the Amira Viewer window and press the *Print Screen* button on the keyboard, whereupon Acrobat will display a popup asking if the user wants to enable capturing from Amira.
- Click ‘yes’ and restart both Amira and Acrobat.
- Reload and display the simple surface again in Amira, select the Amira Viewer window and initiate a capture once more by pressing *Print Screen*.
- In Acrobat; in the 3D Capture Settings (which only appears the first time);
    - Verify that *Enable 3D Capture for this application* is checked.
    - Under the OpenGL tab, set the *Capture units* to those used in Amira. (Unfortunately, it seems that micrometers are not properly shown as units in Acrobat, but an update might solve this).

- In the Acrobat 3D Conversion dialogue;
    - Under the *General* tab;
        - Set *Default Background Color* to white (or another suitable flavour).
        - Set *Default Lighting to 'C*AD Optimized lights'.
        - Set *Default Rendering Style* to 'Solid'.
        - Set *Default Animation Style* to 'None'.
        - Uncheck *Add default views.*
    - Under the *Document* tab,
        - Set the *Page Setup* to A4, Landscape
    - Under the *Optimize* tab,
        - Set the *3D format in PDF* to 'U3D ECMA 3'.
        - Set *U3D: Mesh Quality* to 50%
    - Save these *3D Conversion Settings* as default presets for capture from Amira by clicking the '+' button. (e.g. Amira_capture)

Load surface file

Your object should be a 3D surface, stored by Amira as an Amira Surface file (.surf). As a rule of thumb; keep the complexity of the object below 100,000 triangles. If you are not familiar with surface simplification, consult the Amira documentation.

- Load your object into Amira,
- Connect a *SurfaceView* to the object.

Prepare the surfaceviewers

The object properties of Amira are problematic, as each structure is dual layered (i.e. has an inner and outer surface, each with its own color) and intersections between structures are also exported as separate surfaces. This has been solved using a custom script which enables proper export of the object. For each structure of the object this script creates a separate surface, makes the inside color equal to the outside colour, connects a separate *SurfaceViewer* with correct render settings and renames the surfaces and viewers with correct structure names and numbers.

- Verify that the *SurfaceView* connected to the object is named SurfaceView.
- Load (or drag and drop) the script s*urf2pdf.hx* (see supplementary scripts) into Amira.
- Hide -or remove- the structures you do not wish to export.

3D capture

- Ensure that Acrobat is running.
- Ensure that the Amira Viewer is the active window (i.e. selected).
- Press the *Print Screen* button on the keyboard.
- In Acrobat; accept the conversion dialog using the preset for Amira (e.g. Amira_capture), which generates a pdf of your 3D object.

Rename structures

The structures need to be renamed as the names assigned to the structures in Amira are lost by the capture procedure and reappear in Acrobat as numbers.

- Right click on the object in Acrobat and select *Edit in 3D Reviewer*, which will start the program 'Adobe 3D Reviewer' with the object loaded.
- On the left panel, expand the object tree till the structure names are shown. These will be named M0, M1..Mn and are under *Models*.

- Rename each structure by right clicking on its tree entry and selecting *Rename*. By placing the Amira window adjacent to the Adobe 3D reviewer one can easily refer to the original names.
- Save the changes by selecting *Save back in PDF* from the File menu of the Adobe 3D Reviewer.
- Close Adobe 3D Reviewer and click 'cancel' in its *save changes* dialog.

Save the 3D pdf

- In Acrobat; save the resulting file (by default called LastCapture_<timestamp>.pdf) under an appropriate new name.

The pdf contains a 3D object and can be opened and viewed in Adobe Reader, the freeware 'read-only' program for Adobe's portable document format (pdf). The 3D object can now easily be shared with others, but make sure that their version of Adobe Reader is up-to-date.

Interaction tools

For the resulting 3D-pdf, interaction with the object can be done with the default tools implemented in Acrobat and Adobe Reader. By default the object is shown as a static image. Clicking on the object will activate the interaction with the object and also show the 3D toolbar with the interaction tools (Fig. S1, panel e). These interaction tools enable rotation and resizing of the 3D object. If the 3D toolbar is hidden, right-clicking on the object and selecting *Tools>Show Toolbar* will uncover it. Furthermore, right-clicking on the object and selecting *Show Model Tree* or clicking the button (Fig. S1, button 6) will show the model tree containing the structures of the object (Fig. S1, g). The tree can be expanded by clicking on the plus symbol and structures therein can be made (in)visible by toggling the checkboxes.

ADVANCED STEPS

The above basic protocol can be used to generate a simple 3D-pdf of a reconstructed 3D object. To obtain an advanced 3D-pdf (Fig. S3) enhancements need to be applied to this basic 3D-pdf. These enhancements are independent of each other, so a single (or multiple) modification -as described below- can be applied, whatever suits the document or paper at hand. The main purpose of these enhancements is to present the reader a polished document which is easily and intuitively used. By adding annotations and prepared views, the disseminator can better convey the insights obtained from the object. Also, functionality and robustness can be added by applying scripts to buttons and the document itself.

Views and annotations

Adding views and annotations comes highly recommended, as these are the simplest steps of the advanced additions and allow the disseminator to better convey the insights obtained from the object.

- Views can be added and managed in the model tree (Fig. S1, panel g). The model tree can be opened in multiple ways; right clicking on the activated object and selecting the *Show Model Tree* or pressing on the *Show Model Tree* button (Fig. S1, button 6)
- Position the object interactively into a required view
- Once a suitable view has been set up, pressing the *Create View* button (Fig. S1, button 8) will add a view to the Model Tree. The View panel also contains buttons to delete views, or to set a default view (i.e. the view that is set upon opening the document).

Adding annotations to the object is possible -as of Acrobat version 9.1-. This annotation of structures should not be confused with adding annotations to the document (e.g. *sticky note, callout tool, Text Box tool).* Annotating the 3D object is just to identify structures.

- Add Annotations using the *Add 3D Comment* option of the 3D toolbar. By default, the toolbar is shown at the top of the object, or you can select these tools by right clicking on the object and selecting *tools*
- You can set the colour of the comments in (Edit > Preferences > Measuring (3D) > 3D Measuring Line Colour).

Layout

To apply or create the content for your layout, the vector graphics editor Illustrator is the ideal application to use, as vector graphics keep the file size to a minimum. Furthermore, Illustrator belongs to the same software suite which avoids compatibility problems. The first thing to consider is what the appropriate size and ratio for you document should be. This is dependent of your end product, e.g. A4 when you want to include the object in a standard 'paper' document or a 4:3 ratio if the document is meant to snugly fit a digital presentation, as the majority of displays and beamers use a 4:3 ratio (note that 16:9 is steadily advancing). We recommend to include the following content; a header, a placeholder for the 3D object and buttons to either interact with the 3D object or to serve as hyperlinks to the accompanying paper and affiliated institute. We recommend to design the buttons yourself, as opposed of creating them in Acrobat because this keeps design separated from function. The functional part of hyperlinks and buttons is discussed in the next chapter.
Start by creating a new file in Illustrator: *File > New* and set the page to its appropriate size and aspect ratio.

- Create the minimum graphics;
    - A clear header with a description of the object(s).
    - A simple rectangle which will function as a placeholder for the 3D object.
    - Graphics for buttons and hyperlinks to the original paper and your department's or group's website.
- Once finished with the design, save the file as a design-pdf.
    - Set the *Adobe PDF Preset* in the save dialogue to *Smallest File Size*, which will decrease the file size and remove the layer structure of Illustrator which is currently not fully compatible with 3D objects.

After having created this design-pdf, the 3D object from the basic-pdf created with the *basic protocol* needs to be pasted into this design;

- Open both the basic 3D-pdf and the design-pdf files in Acrobat
- Use the *Select Object Tool* (Fig. S1, button 2) under the *Advanced Editing Tools* to select and copy (ctrl+c) the 3D object
- Paste (ctrl+v) the 3D object into the design pdf.
- Position the object into your placeholder. Positioning can be improved by turning off *Snap to Grid* or by using guides.
- Minor changes to aspects of the design or fields of text can be done using the *TouchUp Object* tool (Fig. S1, button 4) under the *Advanced Editing Tools*. Using this tool, items can be selected and nudged or moved into the right direction.
- Major changes can be achieved, by right-clicking the selected objects and selecting *Edit Object* to open Illustrator which allows more degrees of freedom in editing. Saving the edited item will automatically save the changes back into the 3D-pdf opened in Acrobat.
- Save the resulting 3D-pdf.

The 3D-pdf now contains your custom graphics from Illustrator and a 3D object from Acrobat.

Note: Never load the 3D-pdf in its entirety into Illustrator because Illustrator is not able to correctly handle and save the embedded 3D object within, which will be lost.

Buttons and hyperlinks

The functionality of the 3D-pdfs can be enhanced with buttons. Storing specific views and/or hiding specific structures can also be achieved using the default interaction tools. However, to enhance and simplify interacting with the objects, buttons with similar -or more complex- functionality can be created and behaviour can be added by scripts.

- Using Illustrator, add the appropriate button graphics to your document as described in the *layout* chapter.
- Using Acrobat, overlay these graphics with buttons using the *Button Tool* (Fig. S1, button 3). The buttons can then be edited using the *Button Tool* or the *Select Object Tool* (Fig. S1, button 2). Both tools can also be found under *Tools > Advanced Editing* in the menu.
- Assign a proper name to each button, either by right clicking on it using the *Button Tool* or use the *Select Object Tool* and select *Properties* or *Rename Field.*
- Go to the *Appearance* tab in *Properties* (see above) and hide the superimposed buttons by setting their colour to 'none'.

After setting up your buttons, assign functionality to them, either by what is provided by Acrobat or assign custom scripts to them (see *Scripting* chapter below).

- To make a button link to a website, i.e. hyperlink;
    - Select the button using the *Select Object* tool or *Button* tool
    - Right-click the button and select *Properties*, then the *Actions* tab
        - Set the *Select Trigger* to *Mouse up*
        - Set *Select Action* to *Open a web link* and press *Add...*
        - Enter the URL / link to the site

Scripting

Scripting is the most complicated part of the enhancements, and although the examples provided in this paper are not very complicated, programming experience is highly recommended. Documentation on how to create scripts can be found at http://www.adobe.com/devnet/acrobat/javascript.html and code regarding Javascript and Acrobat 3d can be found at http://www.adobe.com/devnet/acrobat/javascript_acrobt_3d.html. In a 3D-pdf containing a 3D object, JavaScript can be applied at two levels; the document level and the object level. The only piece of script we applied at the object level in the example pdf is the *disable select* script. By default, clicking on a structure selects it and renders it in red, which is unwanted, as we use colours to convey the different types of structures in the object. The 'disable select' script will remove this behaviour from all structures in the object. The other script will be applied at the document level, which allows this script to be called (i.e. used) by all buttons. An overview of every script at the document level can be found on the Acrobat menu *Advanced > Document Processing > Edit all Javascript*. In our published documents, we mainly used functions to hide, show and make structures transparent. The implementation of these functions to buttons is what we will describe here. By using functions, you will centralize the functionality of your document which makes it easier to change this functionality or duplicate it. Furthermore, the functions (if properly coded) are page independent which is very important as it will ensure that the buttons will keep working irrespective of changes in page numbering / order. (e.g. by appending the 3D-pdf to a publication or after combining multiple 3D-pdf files into one pdf)

- To prevent the default select behaviour, load the *disable_select.js* file into the object.
    - Edit the 3D object properties by right-clicking on the object and select *Properties*
        - under the 3D tab select browse to load the script.
- To add the custom functions go to *Advanced > Document Processing > Document JavaScripts*
    - enter a new script name, e.g. 'custom_functions', push the *Add* button, and replace the existing (empty) function with the supplemented code (see custom_functions.txt in the supplementary scripts).
- The functions pasted in the document script can now be called by using the buttons (or any other event). To apply a function to a button:
    - Select the button using the *Select Object* tool or *Button* tool
    - Right-click the button and select *Properties*, then the *Actions* tab
        - Set the *Select Trigger* to *Mouse up*
        - Set *Select Action* to *Run a Javascript* and press *Add...*
        - In the Javascript Editor type the call to the appropriate function (or write custom code), such as **hide("Lumen");** to hide the structure named Lumen. Make sure that the names of the structures in the model tree match those written within quotation marks (note that Javascript is case-sensitive).

Cross sections

A SurfaceCut can be used to clip a part of the object in Amira. This clipping cannot be transferred to Acrobat, but after preparation in Amira, the SurfaceCut(s) can be exported to Acrobat and visualized in the 3D-pdf, even with clipping.

- Load the .surf file into Amira, connect a *SurfaceView*
- Follow the steps *Prepare the surfaceviewers* from the basic protocol
- Connect one (or more) *SurfaceCut(s)* to the .surf file and choose the desired orientation
- Follow the steps *3D Capture* from the basic protocol

The structures and cuts should now be present in a pdf. Check this in the structure tree. The structures and cuts should be separate groups. In rare cases, the model tree is built incorrectly, so verify this. If it is not correct; close and delete the pdf file. In Amira; remove the SurfaceCut(s), hide and subsequently unhide the structures, recreate the SurfaceCut(s) and repeat the *3D capture* step.

When all structures and cuts are present in the captured pdf, it can be further processed in Acrobat.

- Write down the numeric codes of the structures and cuts (M0-Mn) and the structure names to which they should be renamed
- Follow the steps *Rename structures* from the basic protocol
- The structure names of the first cut should be renamed to <structurename>_cut1 for the supplied 'custom_functions' script to work correctly. Repeat this for every additional cut, while incrementing the number (i.e. the second cut should be renamed to <structurename>_cut2)
- Edit the "custom_functions" script (see *Scripting* step) to change the number of cuts (noc) variable (e.g. 2 cuts; change "noc = 0" to "noc = 2")

- Emulate the clipping of SurfaceCuts in Acrobat by;
    - Hide everything (uncheck in the model tree)
    - Unhide the cut you wish to show and rotate it so that the cut is perpendicular to your line of sight.
    - Click on the *Toggle Cross Section* button (Fig. S1, button 9)
    - Click on the *Cross section properties* (the small black triangle next to the *Cross Section* button)
    - In the cross section Properties
        - Check *Enable Cross Section*
        - Uncheck *Show Intersections*
        - Uncheck *Show Cutting Plane*
        - Click on *Align to 3 Points* to align the orientation of the cross section of Acrobat with the cut produced in Amira. For this you need to select three points on the visible structures of this cut
        - unhide the object structures (check in the model tree)
        - (un)check flip, depending on the half you wish to show
        - Nudge the Offset to +0.1% or -0.1%, depending on the direction of the cut. Sometimes the angles are correctly calculated, but the offset is not. Try and nudge the offset dragger until you find the correct spot.
        - Drag the cross section Properties interface to the side (or close it)
    - Unhide the structures you wish to show and / or hide the structures within the cut
    - Press on the *Create View* button to store this viewpoint, make sure that the *Camera Properties, Cross section information* and *Node visibility* are checked
- Repeat the above step for each cut (or inverse cut)
- Link the views to buttons as described in the *scripting* part

Supplementary scripts (see supplementary_scripts.zip)

- surf2pdf.hx
- disable_select.js
- custom_functions.txt