Design and Generation of Interactive 3D PDFs for Medical Data Analysis

Image Guided Therapy Lab 2015

Author: Martin Pletscher

Supervisor: Marius Schwalbe

26.05.2015

Table of Contents

Introduction

Today 3D medical data is rarely used in clinics. To foster the use of 3D medical data a convenient and cost effective way has to be available to clinicians and engineers. Today the exchange of 3D medical data depends mostly on proprietary software, which requires the user to have expensive licenses. Therefore it would be more cost effective if an open and free file format could be used to share 3D medical data.

The Adobe PDF format does support the embedding of 3D model data since version 7. And Adobe Reader is freely available and installed on most workplaces all over the world. Therefore PDF documents seem to be best suited to provide the demanded solution.

On the basis of the previously stated assumptions the following question emerges: Is it possible to generate and embed 3D model data into a PDF document working only with open source software in a fully automated process?

The first step to answer the question was to look at existing solutions. There are mainly tools available to create 3D PDF documents from technical and architectural 3D data. A small set of CAD applications allow the user to create 3D PDF documents directly from within the application. Namely Dassault Solidworks [1] and Catia [2] provide this functionality. Another well establishes solution is the 3D PDF Creator from Tetra4D [3], which provides plugins for many CAD applications in the mechanical and architectural sector.

The only approach in the medical sector was a research project by Axel Newe [4] 2014 which had the goal to enable MeVisLab [5] to export U3D scenes. The project was a success and MeVisLab provides now the functionality to create U3D scenes. These can then be embedded into a 3D PDF document manually. To be able to create U3D scenes automatically, without any user interaction, a license for MeVisLab is required. Therefore no solution for medical applications exists yet. Not to speak of a solution that uses open source software only.

The aim of this work is to develop a pipeline that converts a set of 3D models into a 3D PDF document and to design an interactive user interface which allows clinicians to work with the 3D medical data.

In this report a fully functional pipeline and the iterative development path is shown.

Materials and Methods

Hypothesis 1 – MeVisLab

Starting out from where Axel Newe stopped, the idea emerged to automate the U3D creation process with MeVisLab. In figure 1 an overview of the proposed pipeline is displayed. It is possible to automate MeVisLab with different scripting languages like Python, ML script and JavaScript. But after a short research it was clear that it is not possible to automate MeVisLab without a license. Therefore this idea was not realisable without creating a contradiction to the research question.

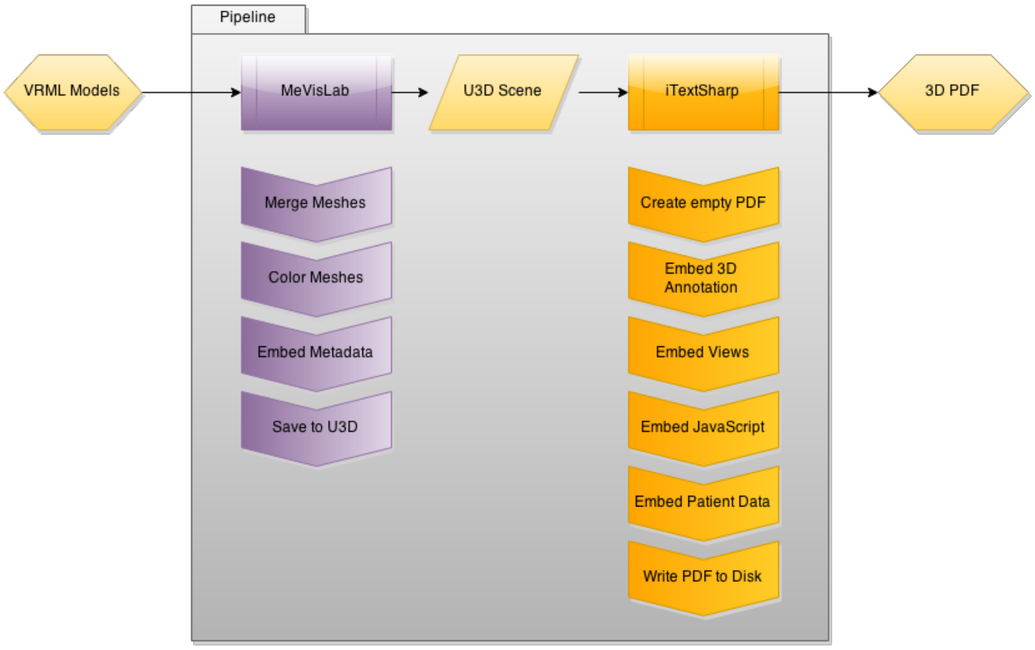


Figure 1: The first proposed pipeline using MeVisLab to create U3D scenes.

Hypothesis 2 – Meshlab

The next pipeline (see figure 2) was then proposed to use Meshlab as the tool to handle the conversion of 3D models to a U3D scene. Meshlab is in theory able to fulfil this task but failed at certain steps. The first problem was that no metadata could be added to the created U3D Scene. And the second even more major problem was that the created U3D scenes did not work with Adobe Reader.

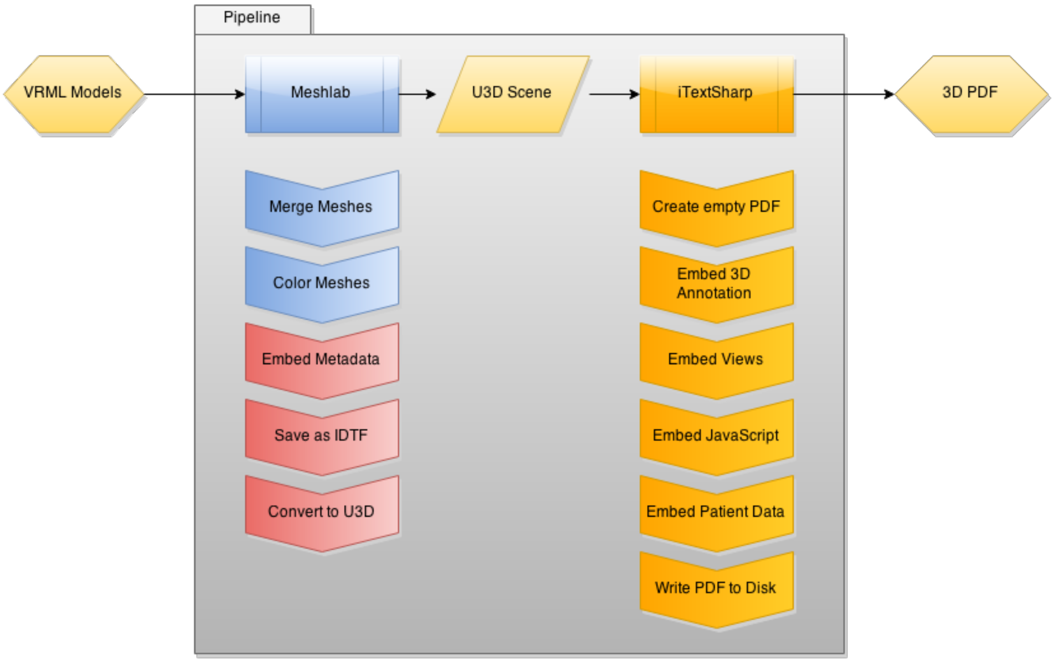


Figure 2: The second proposed pipeline using Meshlab to create U3D scenes.

Hypothesis 3 – The final solution using Octave

The final solution was found by adding Octave into the pipeline. The input of the final pipeline displayed in figure 3 is a set of 3D models existing in the VRML format. At the end the output is a single 3D PDF document consisting of a 3D model and an interactive user interface.

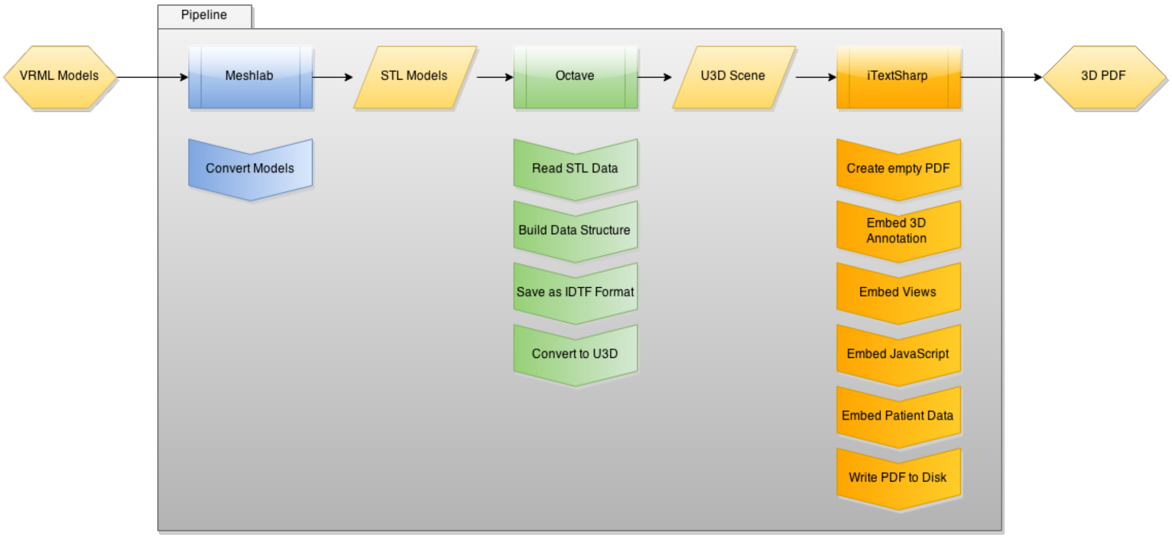


Figure 3: This image shows the final pipeline, the input and output data and the process steps that are executed within the different applications.

The pipeline consists of three software packages: Meshlab [6], Octave [7] and the iTextSharp [8] framework. Each of this software packages is needed to execute a distinctive set of operations to process the 3D models and generate the 3D PDF output.

Converting VRML to STL with Meshlab

The final pipeline uses Meshlab merely to convert VRML models to the STL format. This is done in the first step by calling the command line interface with the input model and the path to the output file. The conversion with Meshlab has proven to work very reliable and provides a wide variety of formats that can be converted to the STL format.

Generating an IDTF Scene with Octave

In the second step Octave is used to create an IDTF scene from the set of 3D Models provided. The Binary STL Format is red by a script found on Matlabcentral [9]. The script reads the model data and the output consists of vertices, faces and normals. With the data of all models a structure is generated that reflects the IDTF format that is used to create the U3D scene.

Before the IDTF structure is generated the models can be optimized by a script [10] which removes double vertices. This can save up to 50% of file size on not optimized models.

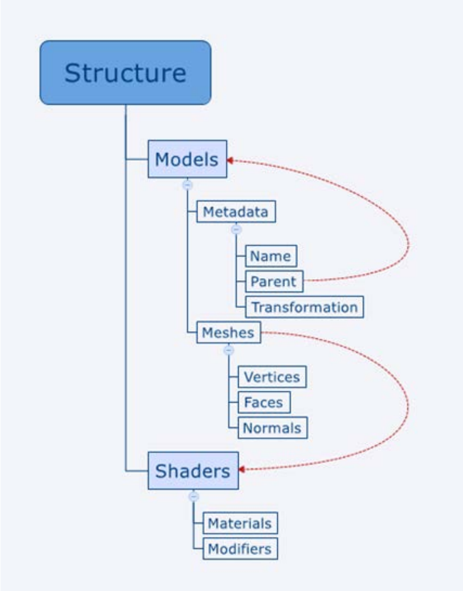


Figure 4: The structure of an IDTF scene. Models hold a set of metadata and one or more mesh. The models are arranged in a hierarchical list.

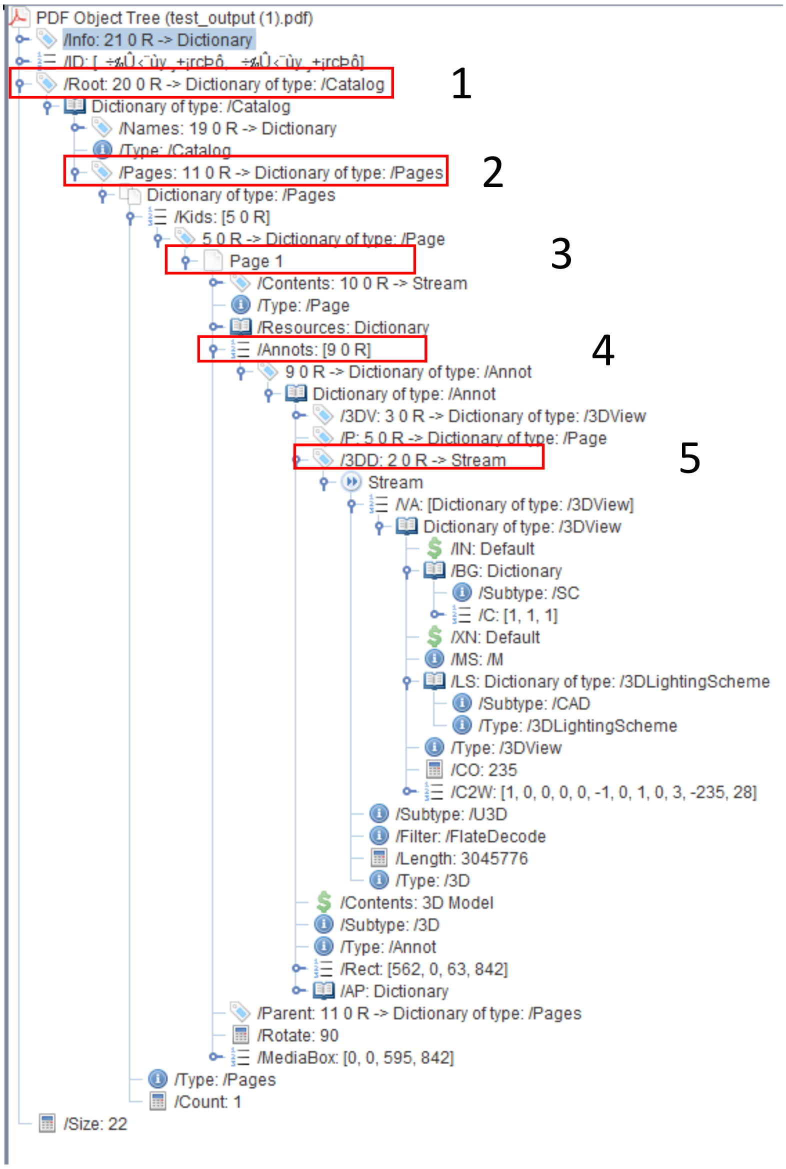
The data of the models is stored in a hierarchical list. As displayed in figure 4 each model includes the metadata and a reference to one parent model. Each model then has a reference to a shader. The shader is required to calculate the lighting at render time. Each shader can include one or more materials and as many modifiers as required. The color information for each model is stored in the shader as a set of RGB values.

Converting IDTF to U3D

To convert the IDTF scene to the U3D format the software IDTFConverter [11] is used. This software was initially developed by Intel and was later released as an open source project. To create a U3D scene the IDTFConverter is called via the command line interface. It creates a U3D file.

Creating a 3D PDF File with iTextSharp

To create the PDF and embed the U3D scene iTextSharp is used. iText is a very mature framework to handle PDF generation and also allows to modify existing documents. It was originally developed in Java and later ported to C#. iTextSharp provides the user with a toolset to create a dictionary structure which represents the data structure of a PDF document.



```
PDF Object Tree (test_output (1).pdf)
/Info: 21 0 R -> Dictionary
/ID: [ ÷‰Û<¨ùy¸+¡rcÞô¸ ÷‰Û<¨ùy¸+¡rcÞô]
/Root: 20 0 R -> Dictionary of type: /Catalog
  Dictionary of type: /Catalog
    /Names: 19 0 R -> Dictionary
    /Type: /Catalog
    /Pages: 11 0 R -> Dictionary of type: /Pages
      Dictionary of type: /Pages
        /Kids: [5 0 R]
          5 0 R -> Dictionary of type: /Page
            Page 1
              /Contents: 10 0 R -> Stream
              /Type: /Page
              /Resources: Dictionary
              /Annots: [9 0 R]
                9 0 R -> Dictionary of type: /Annot
                  Dictionary of type: /Annot
                    /3DV: 3 0 R -> Dictionary of type: /3DView
                    /P: 5 0 R -> Dictionary of type: /Page
                    /3DD: 2 0 R -> Stream
                      Stream
                        /VA: [Dictionary of type: /3DView]
                          Dictionary of type: /3DView
                            /IN: Default
                            /BG: Dictionary
                              /Subtype: /SC
                              /C: [1, 1, 1]
                            /XN: Default
                            /MS: /M
                            /LS: Dictionary of type: /3DLightingScheme
                              /Subtype: /CAD
                              /Type: /3DLightingScheme
                            /Type: /3DView
                            /CO: 235
                            /C2W: [1, 0, 0, 0, 0, -1, 0, 1, 0, 3, -235, 28]
                        /Subtype: /U3D
                        /Filter: /FlateDecode
                        /Length: 3045776
                        /Type: /3D
                    /Contents: 3D Model
                    /Subtype: /3D
                    /Type: /Annot
                    /Rect: [562, 0, 63, 842]
                    /AP: Dictionary
              /Parent: 11 0 R -> Dictionary of type: /Pages
              /Rotate: 90
              /MediaBox: [0, 0, 595, 842]
        /Type: /Pages
        /Count: 1
/Size: 22
```

Figure 5: Dictionary structure of a PDF document. Highlighted elements enumerated from top to bottom: 1: root node 2: page dictionary 3: page 1 4: annotations dictionary 5: U3D scene embedded in a PDF annotation.

In the first step a root dictionary (figure 5:1) is created and a page dictionary (figure 5:2) is added.

A 3D object is handled as an annotation in the PDF document. Therefore a dictionary with the U3D scene object (figure 5:5) is created and attached to the annotations dictionary (figure 5:4) on the first page (figure 5:3) of the PDF document. In a second step a template PDF document is loaded and the design elements are extracted and attached to the page dictionary. Furthermore other required objects to create the user interface are then attached to various locations in the document. The most important object is the JavaScript routine that is directly added to the page dictionary. On its execution during the opening of the document the user interface will be created dynamically.

Results

The result of this work is a compact application that includes the 3D PDF creation pipeline. With this application the user can create 3D PDF documents with a minimum of four clicks. The application has been tested on 3 different Windows systems: Windows 10 Preview, Windows 8.1 and Windows 7. The pipeline works with Octave 3.8.6 and also with the upcoming version 4.0.0. On the conversion side Meshlab 1.3.3 works flawlessly but not the upcoming version 1.3.4. To test the usability of the application one test person was asked to generate a 3D PDF document from a different set of 3D models. The time used to generate the 3D PDF document depends on how many models are used and the size of the models. Generally the generation can be done within minutes.

Adobe introduced the embedding of 3D models with version 7 in the year 2009. In this work only version 10 and 11 could be tested. According to the specifications the generated 3D PDF documents can be opened from version 7 on of Adobe Reader.

The user interface is dynamically created on the opening of the PDF document by a JavaScript routine embedded in the main page. All JavaScript methods are supported by Adobe PDF since version 8. This indicates that the user interface could not be used in Adobe Reader version 7.

Discussion

In this work a fully automated pipeline was developed, which can generate PDF documents with embedded 3D medical data and an interactive user interface to help clinicians analyse the data. The pipeline is only depending on open source software. The only software needed by the end user is a more recent version of Adobe Reader which is completely free of charge and already installed on a majority of all workplaces around the world.

One point has to be considered when using Adobe PDF as a way to exchange 3D medical data. Adobe provides only minimal possibilities to protect PDF documents from being modified and shared. The concern is that under this circumstances it is very difficult to protect the privacy of the patient data. From a regulatory viewpoint it could be very difficult to get a certification for the software because each step has to be certified individually. But certification of open source software like Octave and Meshlab would be nearly impossible.

Outlook

In a future project the interactive user interface could be improved further. In the current version the user interface allows to hide and show different parts of the model. However, Adobe Reader would allow the user interface to modify many other parameters like setting lightning schemes, views, colors, transparency, or even allow to add 3D comments.

Using Meshlab as the conversion tool in the first step opens up the possibility to use any format that Meshlab supports as an input to the pipeline. Meshlab supports currently 18 different 3D model formats and more formats could be added in the future.

There are signs that Adobe could drop the U3D support in the future in favour of the far broader established standard COLLADA [12] which is developed and supported by the KHRONOS group [13]. Adobe is already a member of said industry consortium and has removed the U3D support from its other applications. Therefore this transition is probably only a matter of time.

References

[1] Solidworks.com, (2015). Home. [online] Available at: https://www.solidworks.com/ [Accessed 28 May 2015].

[2] 3ds.com, (2015). CATIA - Dassault Systémes. [online] Available at: http://www.3ds.com/products-services/catia/ [Accessed 28 May 2015].

[3] Tetra4d.com, (2015). tetra4D - the source of 3D PDF. [online] Available at: http://www.tetra4d.com/ [Accessed 28 May 2015].

[4] Newe A. (2015) Towards an easier creation of three-dimensional data for embedding into scholarly 3D PDF (Portable Document Format) files. PeerJ 3:e794 https://dx.doi.org/10.7717/peerj.794

[5] Mevislab.de, (2015). MeVisLab: MeVisLab. [online] Available at: http://www.mevislab.de/ [Accessed 28 May 2015].

[6] MeshLab. [online] Meshlab.sourceforge.net. Available at: http://meshlab.sourceforge.net/ [Accessed 28 May 2015].

[7] Gnu.org, (2015). GNU Octave. [online] Available at: http://www.gnu.org/software/octave/ [Accessed 28 May 2015].

[8] Itextpdf.com, (2015). iText, Programmable PDF software | iText Software. [online] Available at: http://itextpdf.com/ [Accessed 28 May 2015].

[9] (stlread.m) (2005) - File Exchange - MATLAB Central. [online] mathworks.com. Available at: http://ch.mathworks.com/matlabcentral/fileexchange/22409-stl-file-reader/content/STLRead/stlread.m [Accessed 28 May 2015].

[10] (patchslim.m) (2011) - File Exchange - MATLAB Central. [online] mathworks.com. Available at: http://www.mathworks.com/matlabcentral/fileexchange/29986-patch-slim--patchslim-m- [Accessed 28 May 2015].

[11] Software, (2013). Universal 3D Sample Software. [online] SourceForge. Available at: http://sourceforge.net/projects/u3d/ [Accessed 28 May 2015].

[12] Khronos.org, (2015). COLLADA - 3D Asset Exchange Schema. [online] Available at: https://www.khronos.org/collada/ [Accessed 28 May 2015].

[13] Khronos.org, (2015). The Khronos Group Inc.. [online] Available at: https://www.khronos.org/ [Accessed 28 May 2015].