Title

Enriching Scientific Publications with Interactive 3D PDF Figures: A Complete Toolbox

Author & Affiliations

1. Corresponding author:
Name: Axel Newe
Affiliation: Chair of Medical Informatics, Friedrich-Alexander-University Erlangen-Nuremberg, Erlangen, Germany
Postal address: Lehrstuhl für Medizinische Informatik, Wetterkreuz 13, 91058 Erlangen, Erlangen, Germany
Affiliation: Medical Applications Team, Methodpark Engineering GmbH, Erlangen, Germany
Postal address: Methodpark Engineering GmbH, Wetterkreuz 19a, 91058 Erlangen, Germany
E-mail address: axel.newe@fau.de

Manuscript Version Info

Current manuscript version: Preprint v2
Previous manuscript version: Preprint v1 (https://peerj.com/manuscripts/7969v1)
New in this version:

- Binary installers for all operating systems (Windows, MacOS, Linux)
- Wizard for creation of customized modules

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

Abstract

Three-dimensional (3D) data of many kinds is produced at an increasing rate throughout all scientific disciplines. The Portable Document Format (PDF) is the de-facto standard for the exchange of electronic documents and allows for embedding three-dimensional models. Therefore, it is a well suited medium for the visualization and the publication of this kind of data.

The generation of corresponding files has been cumbersome so far. This article presents the first release of a software toolbox which integrates the complete workflow for generating 3D model files and ready-to-publish 3D PDF documents for scholarly publications in a consolidated working environment. It can be used out-of-the-box as a simple working tool or as a basis for specifically tailored solutions. A comprehensive documentation, an example project and a project wizard facilitate the customization. It is available royalty-free and for Windows, MacOS and Linux.

Keywords

PDF; 3D-PDF; Portable Document Format; U3D; Universal 3D; Application

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

Introduction

Throughout many scientific disciplines, the availability – and thus the importance – of three-dimensional (3D) data has grown in the recent years. Consequently, this data is often the basis for scientific publications, and in order to avoid a loss of information, the visualization of this data should be 3D whenever possible [**1**]. In contrary to that, almost all contemporary visualization means (paper printouts, computer screens, etc.) only provide a two-dimensional (2D) interface.

The most common workaround for this limitation is to project the 3D data onto the available 2D plane [**2**], which results in the so-called "2.5D visualization" [**3**]. This projection yields two main problems: limited depth perception and objects that occlude each other. A simple but effective solution of these problems is interaction: by changing the projection angle of a 2.5D visualization (i.e., by changing the point of view), depth perception is improved [**3**], and at the same time objects that had previously been occluded (e.g., the backside) can be brought to sight.

A means of application of this simple solution has been available for many years: the Portable Document Format (PDF) from Adobe [**4**]. This file format is the de-facto standard for the exchange of electronic documents and almost every scientific article that is published nowadays is available as PDF – as well as even articles from the middle of the last century (e.g., [**5**]). PDF allows for embedding 3D models and the Adobe Reader [**6**] can be used to display these models interactively.

Nevertheless, this technology seems not to have found broad acceptance among the scientific community until now, although journals encourage authors to use this technology [**7**,**8**]. One reason might be that the creation of the appropriate model files and of the final PDF documents is still cumbersome. Not everything that is technically possible is accepted by those who are expected to embrace the innovation if the application of this innovation is hampered by inconveniences [**9**]. Generally suitable protocols and procedures have been proposed by a number of authors before, but they all required of toolchain of at least three [**10**,**11**] or even four [**12**,**13**] different software applications and up to 22 single steps until the final PDF was created. Furthermore, some of the proposed workflows were limited to a certain operating system (OS) [**12**], required programming skills [**14**] or relied on commercial software [**15**]. Especially the latter might be an important limiting factor which hampers the proliferation of the 3D PDF format in scientific publishing [**2**,**13**].

This article presents a comprehensive and highly integrated software tool for the creation of both the 3D model files (which can be embedded into PDF documents) and the final, ready-to-publish PDF documents with embedded interactive 3D figures. The presented solution is based on MeVisLab, available for all major operating systems (Windows, MacOS and Linux) and requires no commercial license. The source code is available but does not necessarily need to be compiled since binary add-on installers for all platforms are available. A detailed online documentation, an example project and an integrated wizard facilitate re-use and customization.

Background and Related Work

The Portable Document Format

The Portable Document Format is a document description standard [**16**] for the definition of electronic documents independently of the software, the hardware or the operating system that is used for creating or consuming (displaying, printing…) it. A PDF file can comprise all necessary information and all resources to completely describe the layout and the content of an electronic document, including texts, fonts, images and multimedia elements like audio, movies or 3D models. Therefore, it fulfills all requirements for an interactive publication document as proposed by Thoma et. al. [**17**].

Although an ISO standard (ISO 32000-1:2008 [**18**]), the specification is available to the full extent from the original developer Adobe [**19**] and can be used royalty-free.

Embedding 3D Models into PDF

The fifth edition of the PDF specification (PDF version 1.6, [**20**]), published in 2004, was the first to support so-called “3D Artwork” as an embedded multimedia feature. In January 2005, the Acrobat 7 product family provided the first implementation of tools for creating and displaying these 3D models [**21**].

The latest version (PDF version 1.7, [**16**]) supports three types of geometry (meshes, polylines and point clouds), textures, animations, 15 render modes, 11 lighting schemes and several other features. The only 3D file format that is supported by the ISO standard [**18**] is Universal 3D (U3D, see section below). Support for another 3D format (Product Representation Compact, PRC) has been added by Adobe [**22**] and has been proposed to be integrated into the replacement Norm ISO 32000-2 (PDF 2.0). However, this new standard is currently only available as draft version [**23**] and has not yet been adopted.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

Table 1: Number of publications related to 3D PDFs in biomedical sciences since 2008 (not comprehensive).

Year	Number of publications with embedded/supplemental 3D PDF	Number of publications dealing with/mentioning 3D PDF
2005	-	1
2008	1	-
2009	5	4
2010	2	7
2011	7	6
2012	6	5
2013	7	2
2014	21	7
2015 (until end of November)	29	2

Although the first application in scientific context was proposed in November 2005 [**24**] and thus quite soon after this new technology was available, it took three more years before the first applications were really demonstrated in scholarly articles [**15**,**25**,**26**]. Since then, the number of publications that apply PDF 3D technology either in theory or in practice has increased almost every year (Table 1). The most sophisticated implementation so far is the reporting of planning results for liver surgery where the PDF roots are hidden behind a user interface which emulates a stand-alone software application [**27**].

The Universal 3D (U3D) file format

As outlined above, the U3D file format is the only 3D format that is supported by the current ISO specification of PDF. Initially designed as an exchange format for Computer Aided Construction (CAD), it was later standardized by Ecma International (formerly known as European Computer Manufacturers Association, ECMA) as ECMA-363 (Universal 3D File Format). The latest version is the 4th edition from June 2007 [**28**].

U3D is a binary file format that comprises all information to describe a 3D scene graph. A U3D scene consists of an arbitrary number of objects that can be sorted in an object tree. The geometry of each object can be defined as a triangular mesh, a set of lines or a set of points. A proprietary bit encoding algorithm allows for a highly compressed storage of the geometry data. A number of additional features and entities (textures, lighting, views, animations) can be defined; details are described in previously published articles [**29**].

The scholarly publishing company Elsevier invites authors to supplement their articles with 3D models in U3D format [**8**] and many 3D software tools provide the possibility to export in U3D format. However, most of them are commercial software, but open source solutions like MeshLab [**30**] are available as well.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

Creating 3D Model Files and PDF Documents

Although many tools and libraries are available that support the creation of 3D model files and of final PDF documents, the whole process is still cumbersome. The problems are manifold: some tools require programming skills, some do not support features those are of interest for scientific 3D data (like polylines [**2**] and point clouds [**26**]). Operating system platform support is another issue, as well as royalty-free use.

As regards the creation of the 3D model files, most of these problems have been addressed in a previous article [**2**]. The main problem, however, remains the creation of the final PDFs. Specifying the content and (in particular) the layout of a document can be a complex task and is usually the domain of highly specialized word processor software. Figures and supplements for scholarly publications, on the other hand, usually have a specific layout where only the contents of (a limited number of) pre-defined elements vary. There are at least three common elements for a scientific figure: the figure itself, a short caption text and a longer descriptive text. If the figure is intended to be provided as supplemental information file instead of being integrated into the main article text, some additional information is necessary as well: At least a general headline and an optional reference to the main article should be provided. If the document content is modularized to these five key elements (Figure 1), the creation of the PDF itself becomes a rather simple task, because the layout can be pre-defined.

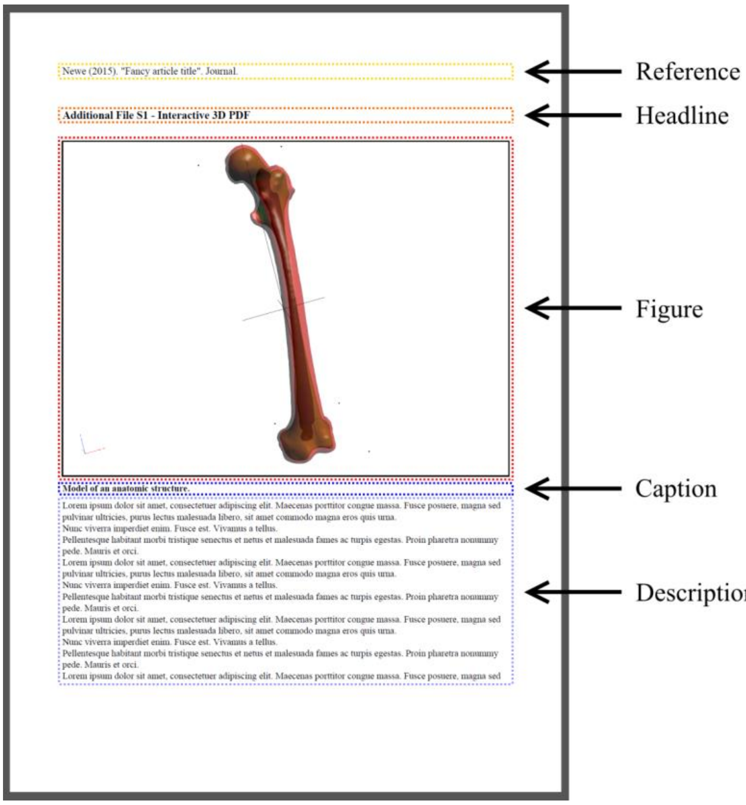

Figure 1: General layout of a scholarly figure if provided as supplemental material.

One last difficulty arises from a peculiarity of interactive 3D figures in PDF: the number viewing options (e.g., camera angle, zoom, lighting…) is nearly unlimited.

Although such a figure is intended to provide all these options, an author usually wants to define an initial view at the objects, if only to simply ensure that all objects are visible. No freely available tool for PDF creation currently provides a feature to pre-define such a view. The movie15 package for LaTeX [**31**] provides a mechanism do determine the view parameters, but that requires the generation of intermediate PDFs.

Finally it must be mentioned that many previously published 3D models are very large – sometimes up to to nearly 100 megabytes [**32**]. In most cases, this size could (and should) be reduced significantly, because the density of polygon meshes does usually not need to be very high for illustrative purposes.

MeVisLab

MeVisLab is a framework for image processing and environment for visual development, published by MeVis Medical Solutions AG and Fraunhofer MEVIS in Bremen, Germany. It is available via download [**33**] for all major platforms (Microsoft Windows, Mac OS and Linux) and has a licensing option which is free for use in non-commercial organizations and research ("MeVisLab SDK Unregistered" license [**34**]). Besides the development features, MeVisLab can be used as a framework for creating sophisticated applications with graphical user interfaces that hide the underlying platform and that can simply be used without any programming knowledge [**35**,**36**,**37**]. MeVisLab has been evaluated as a very good platform for creating application prototypes [**38**], is very well documented [**39**] and supported by an active online community [**40**,**41**].

All algorithms and functions included into MeVisLab are represented and accessed by "modules", which can be arranged and connected to image processing networks or data processing networks on a graphical user interface (GUI) following the visual data-flow development paradigm. By means of so-called "macro modules", these networks can then be converted with little effort into complete applications with an own GUI.

Methods

Elicitation of Requirements

As described above, the generation of the necessary 3D model data and particularly of the final PDF is still subject to a number of difficulties. Therefore, the first step was the creation of a list of requirements specifications with the aim to create a tool that overcomes these known drawbacks.

Two requirements have been identified to be the most important ones: 1) the demand for

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

a tool that creates “ready-to-publish” PDF documents without the need for commercial software and 2) the integration of all necessary steps into a single and easy-to-use interface. Besides these two main requirements, a number of additional requirements have then been identified as well. See Table 2 for a full list of all requirements that were the basis for the following development.

Table 2: Requirements for the development of the software tool. The two main requirements are highlighted in bold font.

ID	Requirement Specification
R1	**The software *shall* create ready-to-publish PDF documents with embedded 3D models.**
R1.1	The software *shall* offer an option to specify the activation mode and the deactivation mode for the 3D models.
R2	**The software *shall* provide an integrated, single-window user interface that comprises all necessary steps.**
R3	The software *shall* be executable under Windows, MacOS and at least one Linux distribution.
R4	The software *shall* be executable without the need to purchase a commerical license.
R5	The software *shall* create 3D model files in U3D format.
R5.1	The software *shall* create view definitions for the 3D model.
R5.2	The software *shall* create poster images for the PDF document.
R6	The software *shall* import mesh geometry from files in OBJ, STL and PLY format.
R6.1	The software *should* import mesh geometry from other file formats as well.
R6.2	The software *shall* offer an option to reduce the number of triangles of imported meshes.
R6.3	The software *shall* offer an option to specify the U3D object name and the color of imported meshes.
R7	The software *shall* import line set geometry from files in text format.
R7.1	The software *shall* offer an option to specify the U3D object name and the color of imported line sets.
R8	The software *shall* import point set geometry from files in text format.
R8.1	The software *shall* offer an option to specify the U3D object name of imported point sets.

Creation of an “App” for MeVisLab

MeVisLab-based solutions presented in previous work [**2**,**29**] already provide the possibility to create U3D files without requiring programming skills and without the need for an intensive training. However, they still needed some basic training as regards assembling the necessary processing chains in MeVisLab. Furthermore, the creation of the final PDF was not possible so far.

Therefore, a new macro module was created for MeVisLab. A macro module encapsulates complex processing networks and can provide an integrated user interface. In this way, the internal processes can be hidden away from the user, who can focus on a streamlined

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

workflow instead. Designed in an appropriate way, a macro module can also be considered as an “app” inside of MeVisLab.

In order to provide the necessary functionality, some auxiliary tool modules (e.g., for the creation of the actual PDF file) needed to be developed as well. Along with the modules for U3D export mentioned above, these auxiliary tool modules were integrated into the internal processing network of the main “app” macro. The technical details of these internal modules are not within the scope of this article. However, the source code is available and interested readers are free to explore the code and to use it for own projects.

The user interface of the app was designed in a way that it guides novice users step-by-step without treating experienced users too condescendingly, though. Finally, a comprehensive documentation including an example project, a wizard for creating tailored PDF modules and a verbose help text was set up.

Deployment of Core Functionality

For the creation of the actual PDF files, version 2.2.0 of the cross-platform, open source library libHaru [**42**] was selected, slightly modified and integrated as third-party contribution into MeVisLab.

Next, the application programming interface (API) of libHaru was wrapped into an abstract base module for MeVisLab in order to provide an easy access to all functions of the library and in order to hide away standard tasks like creating a document or releasing memory. A large number of convenience functions were added to this base module and an exemplary MeVisLab project was set up in order to demonstrate how to use the base module for tailored applications. This base module also served as basis for the PDF creation of the app macro described above. Finally, a project wizard was integrated into the MeVisLab GUI.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

Results

The “Scientific3DFigurePDFApp” module

The new macro module “Scientific3DFigurePDFApp” for MeVisLab provides an integrated user interface for all steps that are necessary for the creation of U3D models files and for the creation of the final PDF documents with embedded 3D models. The app produces U3D models files that are compatible with version 4 of the ECMA-363 standard, poster images in Portable Network Graphics (PNG) format and PDF documents that are compliant with PDF version 1.7 (ISO 32000-1:2008). An example PDF is available as Supplemental File S1.

The user interface is arranged in tabs, whereas each tab comprises all functions for one step of the workflow. By processing the tabs consecutively, the user can assemble and modify 3D models, save them in U3D format, create views and poster images for the PDF document, and finally create the PDF itself step by step (Figure 2).

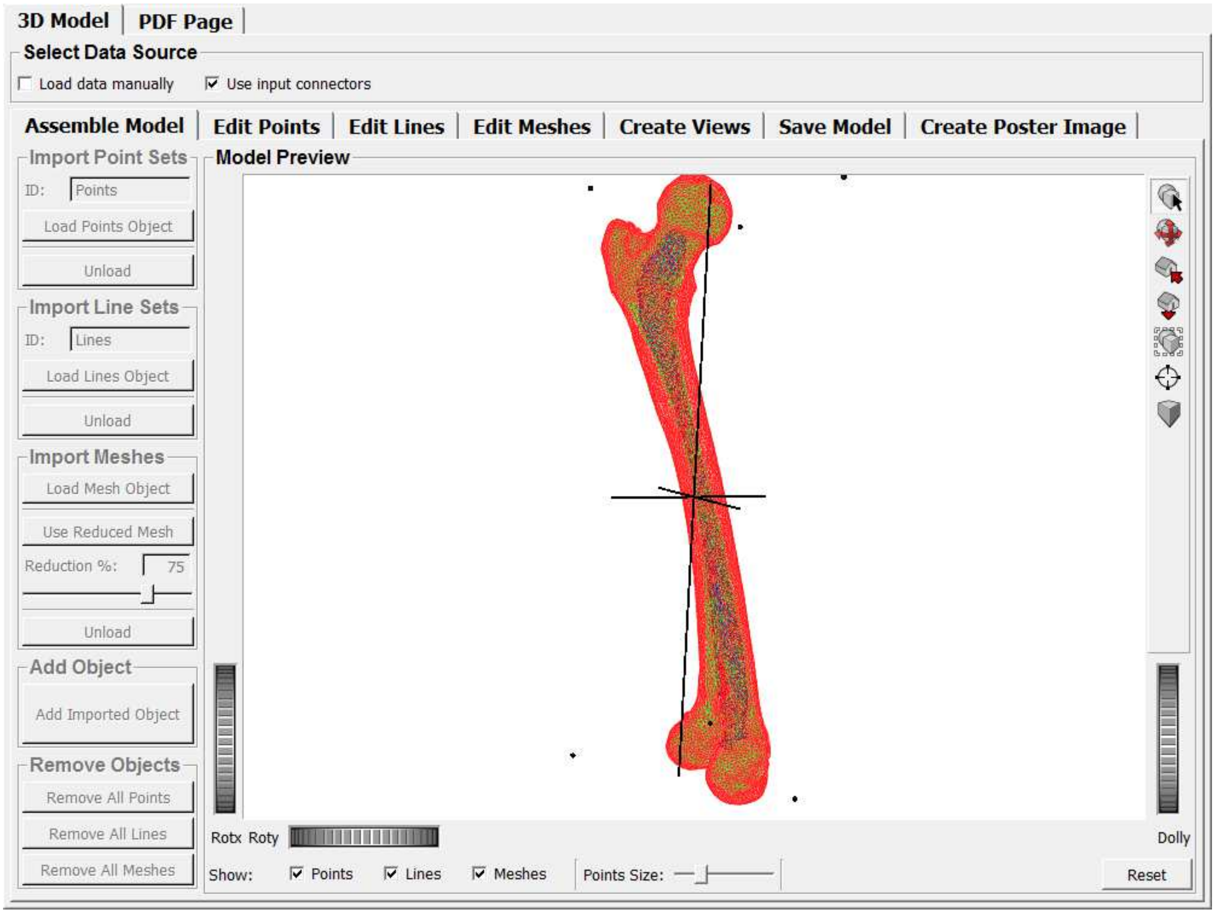

Figure 2: User interface of the app. The user interface comprises all necessary steps for the creation of 3D model files and PDF files. It is arranged in tabs for each step.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

The raw model data can be collected in two ways: either by feeding it to the input connectors or by assembling it by means of the built-in assistant. The former option is intended for experienced MeVisLab users that want to attach the module at the end of a processing chain. The latter option addresses users that simply want to apply the app for converting existing 3D models and for creating an interactive figure for scholar publishing.

The software allows for importing 39 different 3D formats, including point clouds and line sets from files in character-separated value (CSV) format (see Table 3 for a full list).

Objects from different sources can be combined and their U3D properties (color, name, position in the object tree) can be specified. The density of imported meshes can be adjusted interactively and multiple views (i.e., the specification of camera, lighting and render mode) can be pre-defined interactively as well. Finally, it is also possible to create a poster image which can replace an inactive 3D model in the PDF document if the model itself is disabled or if it cannot be displayed for some reason (e.g., because the

Table 3: List of supported 3D formats for import.

File format	Typical File Extension(s)
Stereolithography	*.stl
Stanford Polygon Library	*.ply
Wavefront Object	*.obj
Object File Format	*.off
Blender	*.blend
Raw Triangles	*.raw
Raw Point Clouds	*.csv; *.txt
Raw Line Sets	*.csv; *.txt
3D GameStudio Model	*.mdl
3D GameStudio Terrain	*.hmp
3D Studio Max 3DS	*.3ds
3D Studio Max ASE	*.ase
AC3D	*.ac
AutoCAD DXF	*.dxf
Autodesk DXF	*.dxf
Biovision BVH	*.bvh
CharacterStudio Motion	*.csm
Collada	*.dae; *.xml
DirectX X	*.x
Doom 3	*.md5mesh; *.md5anim; *.md5camera
Irrlicht Mesh	*.irrmesh; *.xml
Irrlicht Scene	*.irr; *.xml
LightWave Model	*.lwo
LightWave Scene	*.lws
Milkshape 3D	*.ms3d
Modo Model	*.lxo
Neutral File Format	*.nff
Ogre	*.mesh.xml, *.skeleton.xml, *.material
Quake I, II, II	*.mdl, *.md2, *.md3
Quake 3 BSP	*.pk3
Quick3D	*.q3o; *q3s
RtCW	*.mdc
Sense8 WorldToolkit	*.nff
Terragen Terrain	*.ter
TrueSpace	*.cob, *.scn
Valve Model	*.smd, *.vta
XGL	*.xgl, *.zgl

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

reading software does not provide the necessary features).

All functions are explained in detail in a comprehensive documentation which can be accessed directly inside MeVisLab. A stand-alone copy of the documentation is available as Supplemental File S2. In order to use the app, it simply needs to be instantiated (via the MeVisLab menu: Modules → PDF → Apps → Scientific3DFigurePDFApp). A full feature list is available in Table 4.

Table 4: List of features.

Category	Features
Data Import	Import external data, import MeVisLab data, import point clouds, import line sets, import meshes from 37 file formats, adjust mesh density, preview import
Point Cloud Editing	Specify point cloud name, specify position in model tree, preview settings
Line Set Editing	Specify line set name, specify position in model tree, specify color, preview settings
Mesh Editing	Specify mesh name, specify position in model tree, specify color, specify opacity, preview settings
View Specification	Specify view name, specify background color, specify lighting scheme, specify render mode, preview settings, specify multiple views
U3D Creation	Store model in U3D format, preview scene
Poster Image Creation	Store poster in PNG format, preview scene, specify superimposed text
PDF Creation	Store document in PDF (v1.7) format, specify header citation text, specify header headline text, specify U3D file, specify poster file, specify model activation mode, specify model deactivation mode, specify toolbar enabling, specify navigation bar enabling, specify animation start mode, specify caption, specify description text

Additional Features for Tailored PDF Creation

The abstract module which wraps the API of the PDF library libHaru into a MeVisLab module was made public (“PDFGenerator” module) and can be used for the development of tailored MeVisLab modules. In order to facilitate the re-use of this abstract base module, an exemplary project was set up (/Projects/PDFExamples/SavePDFTemplate). This project demonstrates how to derive a customized module from the PDFGenerator base module and how to specify the content of the PDF file that will be created by means of the new module. The template code is verbosely annotated and includes examples for setting PDF properties (e.g., meta data, page size, encryption) as well as the document content (including text,

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

images, graphics and 3D models). The output of the SavePDFTemplate module is illustrated in Figure 3.

Finally, a project wizard was integrated into the MeVisLab GUI. It can be accessed via the MeVisLab menu: File → Run Project Wizard… → PDF Module. The wizard consists of three steps (Figure 4) whereof the second step offers the possibility to include demo code which produces the same PDF file as the SavePDFTemplate module described above. The general usage of project wizards in MeVisLab is explained in chapter 23 of the MeVisLab manual (menu: Help → Browse Help Pages → Using MeVisLab → MeVisLab Reference Manual).

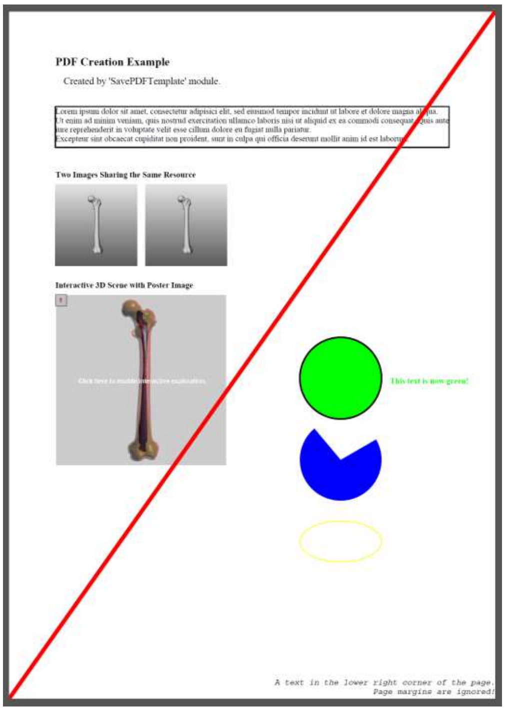

Figure 3: Output of the SavePDFTemplate module.

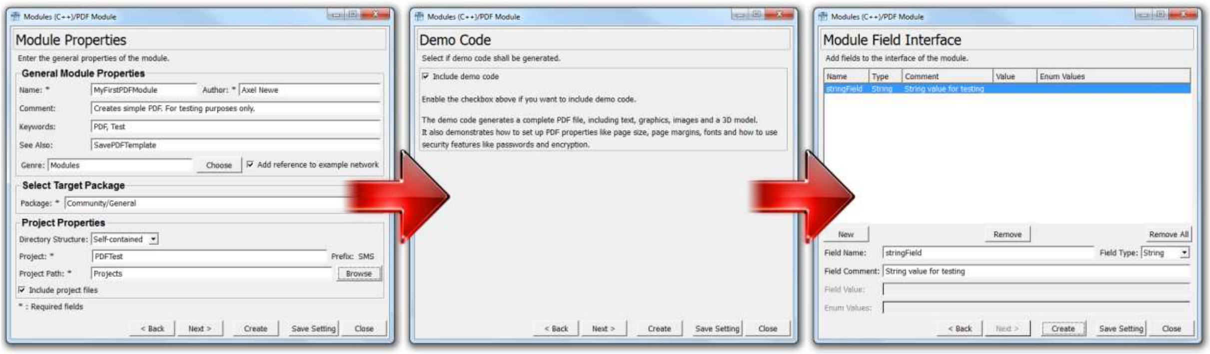

Figure 4: Project wizard for creating customized PDF modules.

Availability

The whole PDF project for MeVisLab (which includes the Scientific3DFigurePDFApp, the PDFGenerator base module, the SavePDFTemplate project, the project wizard and all source code files) is available for Microsoft Windows, MacOS and Linux (tested with Ubuntu 14.04.2). It requires MeVisLab 2.7 or a later version [**33**]. There are two approaches to add the app and the other elements to an existing MeVisLab installation: add-on installers and the online repository of the MeVisLab community sources.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

Installers are self-contained, executable archives that automatically add all necessary files to an existing MeVisLab installation. The target groups for these installers are MeVisLab newcomers and pure users that want to use the Scientific3DFigurePDFApp out-of-the-box. The current version of the installers for all operating systems can be downloaded from the research data repository Zenodo [**43**]. Future updates will be made available via Zenodo as well. A dedicated Zenodo Community Collection named "Three-dimensional Portable Document Format (3D PDF) in Science" has been set up for this purpose [**44**].

All those who are interested in being able to always use the latest version should connect their MeVisLab installation with the community sources which are hosted at GitHub [**41**]. This approach, however, requires compiling the source code and is intended only for experienced users or for users that are willing to become acquainted with MeVisLab.

Discussion

A Toolbox for the Creation of 3D PDFs

The utilization of 3D PDF technology for scholarly publishing has been revealed and proven both useful and necessary by several authors in the past years. The mainstream application of 3D PDF in science, however, is yet to come.

One reason might be the difficult process that has so far been necessary to create appropriate data and relevant electronic documents. This article presents an all-in-one solution for the creation of such files which requires no extraordinary skills. It can be used by low-end users as an out-of-the-box tool as well as a basis for sophisticated tailored solutions for high-end users.

Many typical problems as regards the creation of 3D model files have been addressed and solved. All steps of the workflow are integrated seamlessly. The software is available for all OS platforms and can import and process objects from many popular 3D formats, including polylines and point clouds (Table 3). The density of imported meshes can be adjusted interactively which enables the user to find the best balance between the desired level of detail and the file size.

The main contribution, however, is the possibility to create ready-to-publish PDF documents with a minimum of steps. This approach was proposed to be the ideal solution by Kumar et.al. [**10**]. To best knowledge, this is the first time that such an integrated solution is made available for the scientific community.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

Applications

The areas of application are manifold and not limited to a specific scientific discipline. On the contrary: every field of research that produces three-dimensional data can and should harness this technology in order to get the best out of that data.

One (arbitrary) example for the possible use of mesh models from the recent literature is 3D ultrasound. Dahdouh et.al. recently published about the results of segmentation of obstetric 3D ultrasound images [**45**]. That article contains several figures that project three-dimensional models on the available two-dimensional surface. A presentation in native 3D would have enabled the reader to interactively explore the whole models instead of just one pre-defined snapshot.

Polylines can be used to illustrate nervous fiber tracking. Mitter et.al. [**46**] used 2D projections of association fibers in the fetal brain to visualize their results. A real 3D visualization would have been very helpful in this case as well: While some basic knowledge about a depicted object helps to understand 2D projections of 3D structures, the possibility to preserve at least a little depth perception decreases with an increasing level of abstraction (mesh objects vs. polylines).

This particularly applies to point clouds which can be observed, for example, in an article by Qin et.al. [**47**]: Although these authors added three-dimensional axes to their figure (no. 6) it is still hard to get an impression of depth and therefore of the real position of the points in 3D space.

Suitable Reading Software

The Adobe Reader [**6**] is available free of charge for all major operating systems (MS Windows, Mac OS, Linux). It is currently the only 100% standard compliant software that allows can be used to display embedded 3D models and to let the user interact with them (zooming, panning, rotating, selection of components). However, even the Adobe Reader does not support all U3D features [**48**], e.g., Glyphs and View Nodes.

Experience shows that many users do not expect a PDF document to be interactive. Therefore, possible consumers should be notified that it is possible to interact with the document and they should also be notified that the original Adobe Reader is required for this. Although poster images are a workaround to avoid free areas in PDF readers that are not capable of rendering 3D scenes, missing 3D features of a certain reader could be confusing for a user.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

A Basis for Own Modules

As pointed out in [**2**], the authoring of a PDF document is usually a complex task and thus in most cases it cannot be avoided to separate the generation of 3D model data from the actual PDF authoring. Although the software tool presented in this article mitigates this general problem by integrating model generation and PDF creation, it is still limited to a certain use case and a pre-defined PDF layout.

However, the API of the core PDF functionality is public and designed in a way that facilitates the creation of own PDF export modules. The large number of convenience functions for the abstract base module (PDFGenerator) facilitates the creation of derived modules. These functions massively lighten the programmer's workload by providing a simple access to routine tasks like writing text at a defined position or like embedding a 3D model which would normally require a whole series of API calls. Finally, the built-in wizard generates all necessary project files and source code files to create a fully functional module barebone which only needs to be outfitted with the desired functionality.

Outlook

Although this article represents an important milestone, the development of the PDF project for MeVisLab is ongoing. Future goals are the integration of virtual volume rendering [**14**], animations [**49**] and the parsing of U3D files that have been created with external software. The progress can be tracked via GitHub [**41**] and updates to the binary files will be published regularly.

Conclusion

Three-dimensional data is produced at an increasing rate throughout all scientific disciplines. The Portable Document Format is a well suited medium for the visualization and the publication of this kind of data. With the software presented in this article, the complete workflow for generating 3D model files and 3D PDF documents for scholarly publications can be processed in a consolidated working environment, free of license costs and with all major operating systems. The software addresses novices as well as experienced users: On the one hand, it provides an out-of-the-box solution that can be used like a stand-alone application, and on the other and all sources and APIs are freely available for specifically tailored extensions.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

List of abbreviations

2D – Two-dimensional
3D – Three-dimensional
PDF – Portable Document Format
ISO – International Organization for Standardization
U3D – Universal 3D
PRC – Product Representation Compact
CAD – Computer Aided Construction
ECMA – European Computer Manufacturers Association
GUI – Graphical User Interface
API – Application Programming Interface
PNG – Portable Network Graphics
CSV – Character-separated Value, Comma-separated Value

Competing interest statement

The author declares that he has no competing interests.

Authors' Contributions

Engineered the software: AN. Conceived the study: AN. Drafted and wrote the manuscript: AN.

Acknowledgements

Thanks to Dr. Hans Meine, Fraunhofer MEVIS, Bremen, Germany, for compiling MacOS code and for building installers.
Thanks to Dr. Thomas van de Kamp, Karlsruhe Institute of Technology, Karlsruhe, Germany, for providing test data.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

References

1. Tory M, Möller T. Human Factors in Visualization Research. IEEE Transactions on Visualization and Computer Graphics. 2004; 10(1): 72-84. DOI: 10.1109/TVCG.2004.1260759.

2. Newe A. Towards an easier creation of three-dimensional data for embedding into scholarly 3D PDF (Portable Document Format) files. PeerJ. 2015; 3: e794. DOI 10.7717/peerj.794. PMID: 25780759.

3. Tory M, Möller T. Human Factors in Visualization Research. IEEE Transactions on Visualization and Computer Graphics. 2004; 10(1): 72-84. DOI: 10.1109/TVCG.2004.1260759.

4. Adobe Systems Incorporated. Adobe - PDF: The Global Standard For Trusted, Electronic Documents and Forms. Available: http://www.adobe.com/uk/pdf/. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6S8mi9nMv.

5. Hugh-Jones P. Diabetes in Jamaica. The Lancet. 1955; 266(6896): 891-897.

6. Adobe Systems Incorporated. Adobe - Adobe Reader Download. Available: http://get.adobe.com/reader/otherversions/. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6QkFq9B80.

7. Maunsell J. Announcement Regarding Supplemental Material. J Neurosci. 2010; 30(32): 10599-10600.

8. Elsevier. Interactive U3D models. Available: http://www.elsevier.com/about/content-innovation/u3d-models. Accessed 01 Dec 2015.

9. Hurd JM. The Transformation of Scientific Communication: A Model for 2020. Journal of the American Society for Information Science. 2000; 51(14): 1279-1283. DOI: 10.1002/1097-4571(2000)9999:99993.0.CO;2-1.

10. Kumar P, Ziegler A, Grahn A, Hee CS, Ziegler A. Leaving the structural ivory tower, assisted by interactive 3D PDF. Trends Biochem Sci. 2010; 35(8): 419-422. DOI: 10.1016/j.tibs.2010.03.008. PMID: 20541422.

11. Danz JC, Katsaros C. Three-dimensional portable document format: a simple way to present 3-dimensional data in an electronic publication. Am J Orthod Dentofacial Orthop. 2011; 140(2): 274-276. DOI: 10.1016/j.ajodo.2011.04.010. PMID:21803267.

12. Phelps A, Naeger DM, Marcovici P. Embedding 3D radiology models in portable document format. Am J Roentgenol. 2012; 199(6): 1342-1344. DOI: 10.2214/AJR.12.8716. PMID:23169728.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

13. Lautenschlager S. Palaeontology in the third dimension: a comprehensive guide for the integration of three-dimensional content in publications. Paläontologische Zeitschrift. 2014; 88(1). DOI:10.1007/s12542-013-0184-2..

14. Barnes DG, Vidiassov M, Ruthensteiner B, Fluke CJ, Quayle MR, McHenry CR. Embedding and Publishing Interactive, 3-Dimensional, Scientific Figures in Portable Document Format (PDF) Files. PLoS ONE. 2013; 8(9): e69446. DOI: 10.1371/journal.pone.0069446. PMID:24086243.

15. Ruthensteiner B, Heß M. Embedding 3D Models of Biological Specimens in PDF Publications. Microsc Res Tech. 2008; 71(11): 778-786. DOI: 10.1002/jemt.20618. PMID:18785246.

16. Adobe Systems Incorporated. Document management — Portable document format — Part 1: PDF 1.7. 2008. Available: http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000_2008.pdf. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6S8xkrvPw.

17. Thoma GR, Ford G, Antani S, Demner-Fushman D, Chung M, Simpson M. Interactive Publication: The document as a research tool. Web Semantics. 2010; 8(2-3): 145-150. PMID: 20657757.

18. International Organization for Standardization. ISO 32000-1:2008 Document management -- Portable document format -- Part 1: PDF 1.7. 2008. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51502. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6QkFZgVk9.

19. Adobe Systems Incorporated. PDF Reference and Adobe Extensions to the PDF Specification. Available: http://www.adobe.com/devnet/pdf/pdf_reference.edu.html. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6S8y0ShjG.

20. Adobe Systems Incorporated. PDF Reference, fifth edition, Adobe Portable Document Format, Version 1.6. 2004. Available: http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_archives/PDFReference16.pdf. Accessed 01 Dec 2015.

21. Adobe Systems Incorporated. Adobe Announces Acrobat 7.0 Software Availability. Available: https://www.adobe.com/aboutadobe/pressroom/pressreleases/200501/010505Acrobat.html. Accessed 01 Dec 2015.

22. Adobe Systems Incorporated. Adobe® Supplement to the ISO 32000, BaseVersion: 1.7, ExtensionLevel: 3. 2008c. Available: http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/adobe_supplement_iso32000.pdf. Accessed 01 Dec 2015.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

23. International Organization for Standardization. ISO/DIS 32000-2 Review Guide. 2014. Available: http://www.pdfa.org/wp-content/uploads/2014/09/ISO-32000-2_DIS-1_REVIEW_ONLY.pdf. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6byQJVKAs

24. Zlatanova S, Verbree E. The third dimension in LBS: the steps to go. In: The 3rd symposium on LBS&TeleCartography, 28-30 November. In: Geowissenschaftliche Mitteilungen, Heft Nr. 74; 2005; Vienna, Austria. 185-190.

25. Kumar P, Ziegler A, Ziegler J, Uchanska-Ziegler B, Ziegler A. Grasping molecular structures through publication-integrated 3D models. Trends Biochem Sci. 2008; 33(9): 408-412.

26. Barnes DG, Fluke CJ. Incorporating interactive 3-dimensional graphics in astronomy research papers. New Astron. 2008; 13(8): 599-605.

27. Newe A, Becker L, Schenk A. Application and Evaluation of Interactive 3D PDF for Presenting and Sharing Planning Results for Liver Surgery in Clinical Routine. PLoS ONE. 2014; 9(12): e115697. DOI: 10.1371/journal.pone.0115697.

28. ECMA International. Standard ECMA-363, Universal 3D File Format, 4th edition (June 2007). 2007. Available: http://www.ecma-international.org/cgi-bin/counters/unicounter.pl?name=ECMA-363_4thedition&deliver=http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-363%204th%20edition.pdf. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6QkI44nqn.

29. Newe A, Ganslandt T. Simplified Generation of Biomedical 3D Surface Model Data for Embedding Into 3D Portable Document Format (PDF) Files for Publication and Education. PLoS ONE. 2013; 8(11): e79004. DOI: 10.1371/journal.pone.0079004.

30. MeshLab Developers. MeshLab. Available: http://meshlab.sourceforge.net/. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6S9268uzQ.

31. Grahn A. The movie15 Package. 2005. Available: http://solar.physics.montana.edu/kankel/ph567/resources/LaTeX/graphics-animations/movie15.pdf. Accessed 01 Dec 2015.

32. Krause DW, Wible JR, Hoffmann S, Groenke JR, O'Connor PM, Holloway WL et al. Craniofacial Morphology of Vintana Sertichi (Mammalia, Gondwanatheria) from the Late Cretaceous of Madagascar. Journal of Vertebrate Paleontology. 2014; 34(Supplement 1): 14-109. DOI: 10.1080/02724634.2014.976129.

33. MeVis Medical Solutions AG and Fraunhofer MEVIS. MeVisLab - Download. Available: http://www.mevislab.de/download/. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6SA6T1zzq.

34. MeVis Medical Solutions AG & Fraunhofer MEVIS. MeVisLab - Versions and Licensing. Available: http://www.mevislab.de/mevislab/versions-and-licensing/. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6SA6b5Nv0.

35. Koenig M, Spindler W, Rexilius J, Jomier J, Link F, Peitgen HO. Embedding VTK and ITK into a visual programming and rapid prototyping platform. Proc. Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display. 2006; 6141(1).

36. Heckel F, Schwier M, Peitgen HO. Object oriented application development with MeVisLab and Python. Lecture Notes in Informatics (Informatik 2009: Im Focus das Leben). 2009; 154: 1338-1351.

37. Ritter F, Boskamp T, Homeyer A, Laue H, Schwier M, Link F et al. Medical image analysis. IEEE Pulse. 2011; 2(6): 60-70.

38. Bitter I, Van Uitert R, Wolf I, Ibanez L, Kuhnigk JM. Comparison of four freely available frameworks for image processing and visualization that use ITK. IEEE T Vis Comput Gr. 2007; 13(3): 483-493.

39. MeVis Medical Solutions AG and Fraunhofer MEVIS. MeVisLab - Documentation. Available: http://www.mevislab.de/developer/documentation/. Accessed 01 Dec 2015. Archived at: http://www.webcitation.org/6SADHStrk.

40. MeVis Medical Solutions AG and Fraunhofer MEVIS. MeVisLab - Community. Available: http://www.mevislab.de/developer/community/. Accessed 01 Dec 2015.

41. MeVisLab Community. MeVisLab Community Modules. Available: https://github.com/MeVisLab/communitymodules/tree/master/Community. Accessed 01 Dec 2015.

42. libHaru Developers. libHaru - Free PDF Library. Available: http://libharu.org/. Accessed 01 Dec 2015.

43. Newe A. MeVisLab Community PDF-AddOn (2016-01-26 release). 2016. Available: https://zenodo.org/record/45182. Accessed 26 Jan 2016. DOI: 10.5281/zenodo.45182

44. Newe A. Three-dimensional Portable Document Format (3D PDF) in Science. Available: https://zenodo.org/collection/user-3d-pdf. Accessed 11 Jan 2016.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016

45. Dahdouh S, Angelini ED, Grangé G, Bloch I. Segmentation of embryonic and fetal 3D ultrasound images based on pixel intensity distributions and shape priors. Medical Image Analysis. 2015; 24(1): 255-268. DOI: 10.1016/j.media.2014.12.005. PMID: 25655408.

46. Mitter C, Prayer D, Brugger PC, Weber M, Kasprian G. In Vivo Tractography of Fetal Association. PLoS ONE. 2015; 10(3): e0119536. DOI: 10.1371/journal.pone.0119536. PMID: 25742520.

47. Qin Y, Yao W, Vu TT, Li S, Niu Z, Ban Y. Characterizing Radiometric Attributes of Point Cloud Using a Normalized Reflective Factor Derived From Small Footprint LiDAR Waveform. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2015; 8(2): 740-749. DOI: 10.1109/JSTARS.2014.2354014.

48. Adobe Systems Incorporated. U3D Supported Elements. 2007c. Available: http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/U3DElements.pdf. Accessed 01 Dec 2015.

49. van de Kamp T, dos Santos Rolo T, Vagovic P, Baumbach T, Riedel A. Three-Dimensional Reconstructions Come to Life – Interactive 3D PDF Animations in Functional Morphology. PLoS ONE. 2014; 9(7): e102355. doi:10.1371/journal.pone.0102355.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1594v2 | CC-BY 4.0 Open Access | rec: 27 Jan 2016, publ: 27 Jan 2016