The 10th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2009)
K. Debattista, C. Perlingieri, D. Pitzalis, and S. Spina (Editors)

Publishing 3D Content as PDF in Cultural Heritage

M. Strobl[1] R. Berndt[1] V. Settgast[1] S. Havemann[1] D. W. Fellner[1,2]

[1]Institut für ComputerGraphik & Wissensvisualisierung, TU Graz, Austria
[2]TU Darmstadt & Fraunhofer IGD, Germany

Abstract

Sharing 3D models with embedded annotations and additional information in a generally accessible way still is a major challange. Using 3D technology must become much easier, in particular in areas such as Cultural Heritage, where archeologists, art historians, and museum curators rely on robust, easy to use solutions. Sustainable exchange standards are vital since unlike in industry, no sophisticated PLM or PDM solutions are common in CH. To solve this problem we have examined the PDF file format and developed concepts and software for the exchange of annotated 3D models in a way that is not just comfortable but also sustainable. We show typical use cases for authoring and using PDF documents containing annotated 3D geometry. The resulting workflow is efficient and suitable for experienced users as well as for users working only with standard word processing tools and e-mail clients (plus, currently, Acrobat Pro Extended).

Categories and Subject Descriptors (according to ACM CCS): Graphics Utilities [I.3.4]: Graphics editors— Methodology and Techniques [I.3.6]: Languages,Standards—

1. Introduction

Three-dimensional models can be incredibly rich and useful sources of historical information. 3D models may be obtained by measurements from (typically incomplete) physical remains of historical artifacts, or by synthetic reconstruction, hypothesizing about the original shape and appearance of things in the ancient past. In either case, a 3D model as such is not very useful without semantics, i.e., without interpretation, mark-up, annotation, or highlighting, that help to discern the important from the unimportant. Accurate interpretations rely on accurate raw data; accurate raw data mean nothing without interpretation.

CH professionals such as historians, archeologists, curators, but also conservators and restorers, have a hard time today making use of 3D models as information sources. It is by far more easy to create, annotate, and exchange drawings or photographs than 3D models. However, many questions can *not* be answered from photographs or 2D drawings. Complex spatial configurations, shape details, or distance measurements are barely possible with photographs. Whenever a thing needs to be looked at from all possible sides, and for serious questions beyond aesthetic imaging, e.g., sections or occludedness, 3D models are the medium of choice.

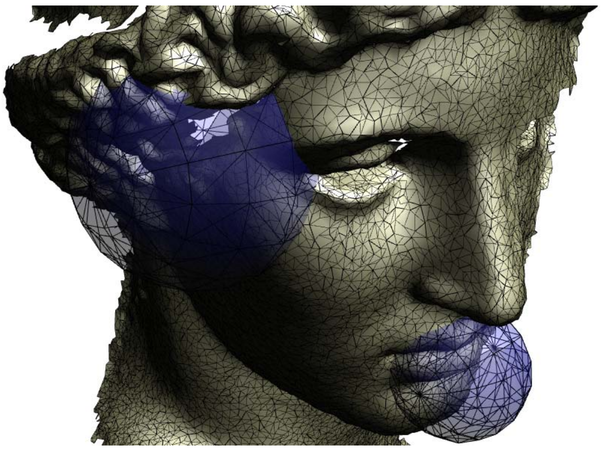

Figure 1: *A 3D model file with embedded annotations. If this paper is viewed in a PDF reader with 3D support, clicking the image activates 3D. Blue spheres have URLs attached.*

DOI: 10.2312/VAST/VAST09/117-124

Research requires communication. Assuming 3D models are available and two CH professionals A and B wish to develop and share hypotheses and insights based on these models, which options do they have? In essence there are two:

- **Common Model Repository:** Both researchers refer to the same web-based information resource, a common 3D infrastructure, e.g., by sending each other links (URLs) that refer to positions in 3D models in the DB.
- **General Purpose Modeling Tool:** Researcher A loads a model into 3D Studio Max, Maya, Blender etc., inserts annotations (typically by creating additional geometry or text embedded in 3D), saves it, and makes it available to researcher B, e.g., via e-mail or ftp.

The repository solution is clearly more sustainable. It makes a sharp distinction between model and annotation, and it permits multiple interpretations; but it relies on an online connection, and on the long-term availability of the 3D model in the repository. The second option has the advantage that it is self-contained, but the annotation follows no standard, and it "pollutes" the 3D model. Technical issues are that import/-export filters of 3D tools greatly vary in quality, and there is always the danger of loss of information (fidelity). So A and B should better agree on using the same 3D software.

Another problem is using 3D models in scientific publications. Today, 3D models appear in publications only as 2D images, e.g., as nice rendering, essentially a digitally born photograph. Nice renderings are difficult to obtain and require being acquainted with rendering methods and their parameters; using, e.g., MentalRay in Maya or 3D Studio Max requires expert knowledge. Furthermore, it seems odd replacing the richer three-dimensional information by its projection to 2D. Why not include the 3D model in the paper, all the more since all scientific articles that are published today are available in electronic form – typically in PDF format.

1.1. Contribution

We present a PDF-based solution that combines the advantages and avoids the disadvantages of the two approaches from the previous section. Since PDF is the de-facto standard for electronic documents, it is only natural to examine its usefulness for reasoning about historical 3D models. Since version 1.6 (from 2005) PDF supports 3D. PDF viewers are available basically everywhere, so no special software installation is required for viewing 3D models embedded in PDF documents. However, currently only the gratis *Acrobat Reader* software from Adobe Inc. fully supports viewing embedded 3D; but since 3D is now part of the PDF standard, other PDF readers will follow sooner or later.

Previous work has shown how sustainable 3D markup can be achieved by a strict separation of 3D model (stored in binary format, e.g., PLY) and annotation, stored as a Collada XML file, referring to the 3D model as external resource [HSBE08]. Our paper builds upon this work and extends the markup facility, so that no special software except a 3D capable PDF viewer is necessary to inspect the annotated 3D models. The contributions of our paper are:

- Explaining the export of annotated 3D models to PDF
- Explanation of the manipulation of the PDF structure
- Distinction between model and annotation is still possible
- Three use cases: Sending annotated 3D models as PDF via e-mail, including them in scientific publications using Microsoft Word, and using LaTeX.

1.2. Benefit

Our solution allows easy inclusion of annotated 3D models into scientific publications, allows archiving them, and allows taking annotated 3D models to places without network connection (e.g., on a remote excavation campaign).

It is important to maintain the difference between model and annotation, which is not possible reliably if the model is annotated using general-purpose 3D software. Our approach is to exploit the built-in annotation capabilities of PDF, i.e., to integrate the markup into the PDF document structure, rather than into the 3D model. The model can be discerned from its interpretation also in the long run, and both can be extracted separately from the PDF. This is an important prerequisite of any 3D file format suitable for archival. Furthermore, in the common infrastructure scenario mentioned above, where a large 3D model repository is available as online resource, the annotated PDF can be regarded as a snapshot that is available even if the online resource vanishes, or if the link location changes. A central CH infrastructure clearly is an extremely desirable (and ambitious) goal, and it is incredibly valuable as research tool; however, the ability to achieve independence from it is desirable as well.

Adobe provides the *Acrobat Pro Extended* software for inserting 3D models into PDF documents, which (in 2009) costs more than 900 Euros. Our current solution depends on this software for exporting (but not for viewing) annotated 3D models as PDF. Clearly, our goal is to become independent from any proprietary software; ways for doing so are sketched in section 6.

2. Previous and Related Work

PDF is used intensively in the manufacturing industry as exchange format for annotated 3D models. Sections and 2D drawings are insufficient for describing free-form surfaces. So PDF export functionality is available in almost all major CAD applications, just to mention Dassault Catia V5, Bentley MicroStation and Graphisoft ArchiCAD. Main reasons for using it are the advantage of 3D models over 2D plans, the wide spread availability of PDF viewers, and security considerations. Commercial solutions like PDF3D from Visual Technology Services [Vis09] allow integrating PDF ex-

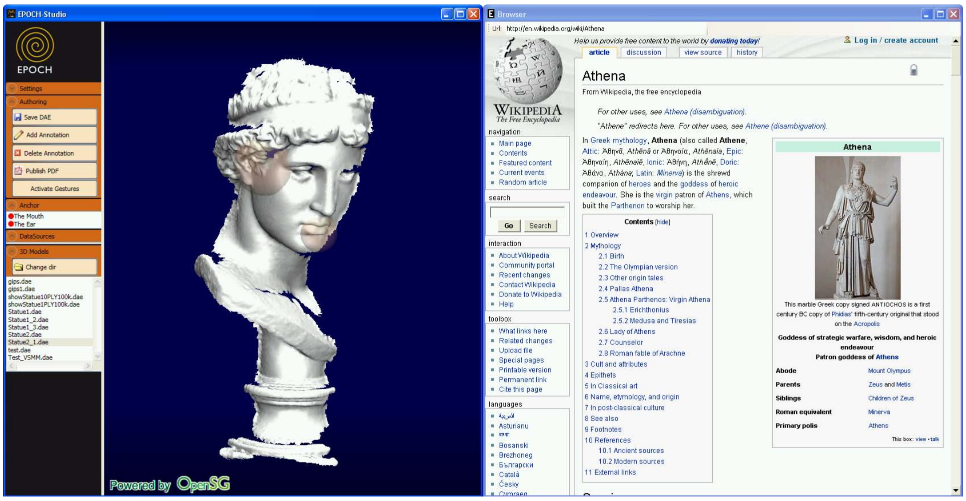

Figure 2: *The 3D model is annotated using the* EpochStudio. *The software is composed of a menu (left), a 3D view for the detailed inspection of the model (middle), and a browser for web-based research (right). Annotations can be placed on spots on the surface of the 3D model, shown as transparent balls. Annotations can be connected to one or multiple URLs.*

port also with existing production work-flows and applications. PDF is used for product presentation and explanation, e.g., for 3D user manuals that explain unambiguously the assembly and disassembly of complex machine parts.

In the manufacturing industry the standard today for organizing huge masses of 3D content with rich structure are extremely complex product data management (PDM) systems. They help developing industrial plants with huge machines composed of various parts and sub-parts, each produced by different companies. PDM systems manage versioning information and help keeping up with change requests and maintaining geometric consistency. Once a facility is built, digital plans and built reality are kept in sync using similarly complex *product lifecycle management* (PLM) solutions. PDM/-PLM systems, for example *Windchill* from PTC, allow creating and exchanging annotations to 3D models, and importing / exporting them via PDF.

It may be surprising that in fact the same requirements apply to 3D in Cultural Heritage (CH): Large excavation sites exhibit extremely complex geometric configurations (strata), on multiple scales (from single artefact to landscape), and both physical state and knowledge/interpretation change over time. But unlike in the industry setting, not just clean CAD data need to be managed, but huge masses of measured 3D data that affected by noise, have outliers, etc. Unfortunately, PDM/PLM solution cost several 100K Euros.

But also free and open source software exists for adding 3D content to a PDF document. The *Meshlab* software by Paolo Cignoni et al. from ISTI-CNR in Pisa, Italy [CCC*08], allows editing 3D meshes and converting many 3D file formats to U3D. The *movie15* package for LaTeX by Alexander Grahn provides means for embedding various multimedia objects into PDF files generated by LaTeX. Amongst other formats *Universal 3D* (U3D) files can be embedded, and Javascript can also be attached to the PDFs' 3D data streams. However, the annotation of a 3D model must still be done using an authoring tool like Adobe Acrobat.

Concerning the PDF standard, PDF/A-1 has been approved in 2005 as the standard for long-term archival of textual PDF documents [ISO05]. The next version PDF/A-2 [ISO08], announced for 2009, will be based on PDF 1.7 and is expected to contain support for multimedia content, especially 3D. This would make PDF indeed the preferential format for the sustainable publishing of 3D content mixed with other data like text, sounds or videos.

Using 3D in PDF has apparently not received much research attention, and is also not common in CH. Rieko Kadobayashi developed a system for automatic 3D blogging in CH that uses annotations to 3D models [Kad05]. Visitors of virtual museum artefacts can make textual annotations and attach them to spots on a 3D model of the artefact. Other users can subscribe the annotations of a particular model as a blog. This system could use our approach to send out the annotations as PDF, or to let uses make a snapshot of the annotations of a particular object.

3D in PDF was used as research tool by Barnes and Fluke in the area of astronomy [BF08]. Their S2PLOT library for visualization, e.g., of astronomical data whose VRML export was complemented by PDF3D export, because the authors noted that 2D illustrations are insufficent to understand the complex 3D data in astronomy. However, they use it only as export filter and make no particular use of annotations.

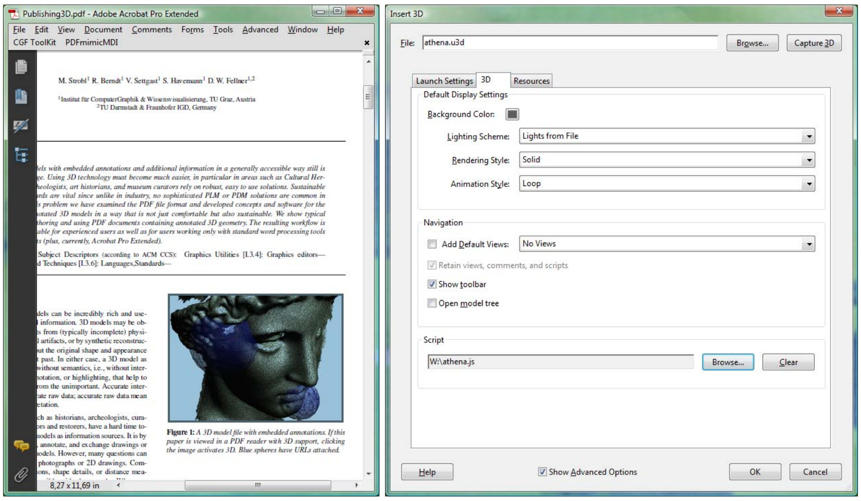

Figure 3: *Using the* 3D Tool *of Acrobat for inserting 3D content.*

3. Use Cases for 3D in PDF

This section describes different options for the workflow in order to publish annotated 3D content:

- Annotating 3D content with EpochStudio
- Adding 3D content using Acrobat 9
- Adding 3D content using Word
- Adding 3D content using LaTeX

The workflow for two of the most popular text authoring tools, Microsoft Word and LaTeX, is described in the following. Note that 3D can still be used even for other text authoring tool: The fallback solution is simply always to generate the document with a placeholder, and to replace this placeholder using *Acrobat Professional* (Extended not needed).

3.1. Authoring 3D annotations with EpochStudio

The annotation of 3D models is done using the *EpochStudio* software that was presented in [HSBE08]. Figure 2 shows a typical authoring session with the 3D view for displaying the model in the middle and a web browser for investigation on the right.

The model is loaded from the local hard drive or from a remote data source. By clicking on the authoring tab on the left, the user can select *"Add Annotation"* from the menu. Annotations are then added by clicking on the surface of the 3D object. A dialog lets the user choose an anchor name. This name can be used later to create a reference to the annotation. Anchors are visualized as spheres in the 3D view and they also appear on the left in the list of anchors.

An arbitrary number of URLs can be attached to each annotation. With the included browser it is possible to research on suitable material without leaving the application. After all annotations are added the user can export a simple Collada file by clicking on *"Save DAE"*. This Collada file holds the scene graph hierarchy and the annotations. The 3D models, however, are not contained in the Collada file but only referenced using a link. To generate a PDF, the user selects another export option, *"Publish PDF"*. A PDF file containing the 3D model as well as the annotations is then written to disk. For the manual integration we also export the 3D scene as VRML and the CSV file separately (see section 4.3).

3.2. Fallback solution: Acrobat Professional

With Acrobat Professional, 3D content can be integrated into an existing PDF document using the *3D Tool* (from Tools / Multimedia). The user specifies the area where the 3D content shall be placed on the page by drawing a rectangle. A file dialog asks for the 3D model, and advanced options include an optional JavaScipt file (e.g. for animations), various render styles, background color, and many more (see Fig. 3).

3.3. PDF with 3D from Word (with PDFMaker)

The installation of Adobe Acrobat includes the Word add-in PDFMaker, which enables advanced PDF features like embedding videos and 3D content. Figure 4 shows the toolbar of the PDFMaker in Microsoft Word 2007. The *"Embed 3D"* button allows drawing a rectangle on the page and asks for the 3D content file to insert. The context menu of the inserted content allows changing the options concerning the JavaScript file (used, e.g., for animations) and other options like render style, background color, etc.

Without installation of the PDFMaker add-in the fallback solution is to insert a placeholder (explained in section 3.2).

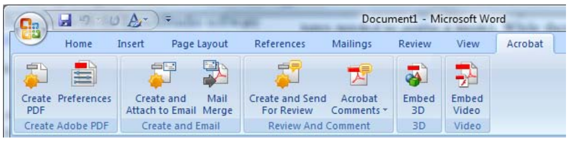

Figure 4: *The* PDFMaker *add-in in Microsoft Word.*

3.4. PDF from LaTeX

It is also possible to include the 3D content into a LaTeX document. The straightforward solution, to use pdflatex to include the generated pdf file directly, leads to an error, since pdflatex can not include PDF files of version greater 1.4 directly. Hopefully this limitation will vanish soon. There are basically two alternative options. The first one uses the *movie15* package to include external media. But it is restricted to the U3D file format, and the drawback is that the event scripting capabilities (using JavaScript) are very limited: External links are not accessible from within the 3D canvas. Hopefully this limitation will vanish in future releases of the Acrobat Reader or other PDF viewers.

For the creation of this paper the fallback solution was used to include a PDF generated by the Epoch PDF Publisher at a placeholder position. The LaTeX document is generated as usual, but at a specific place, a \vspace or a \newpage directive generates empty space. The generated PDF was included using Acrobat Professional.

4. Implementation Details

This section describes the details of the PDF file format, and the different options that exist for including 3D models and annotations, as well as the workflow of our software implementing these options.

4.1. The PDF file structure

The internal structure of a PDF document is defined by the *Carousel Object Structure*, the *COS-tree*, which is conceptually very similar to the DOM tree of an XML file, or the scene graph of a 3D file. Strictly speaking, the COS is not a tree and not even a directed acyclic graph (DAG), but may even be a general graph: The COS is a hierarchy of nodes where each node is a dictionary (list of key-value pairs), where the value can be a number, a string, an array, or again a dictionary. The value can be *any* dictionary, even the same dictionary again, or the parent dictionary - which is used quite frequently in fact (all the P entries). We call this structure COS-tree because the most common way of showing it is in a tree browser. It can be displayed, e.g., using the *Acrobat Professional* software using Advanced / Preflight / Options / Browse Internal PDF structure. Fig. 5 shows the most important nodes of the first levels of the COS-tree of a one-page PDF document with an embedded annotated 3D model created by our software. The tree nodes can be denoted using path expressions such as

Page1.Annots.0.3DD.Binary stream

which is the location of the actual 3D model, encoded in binary form. Table 1 shows the path expressions of the parts of the COS-tree shown in Fig. 5.

(a)	Page1
(b)	Page1.Annots
(c)	Page1.Annots.0
(d)	Page1.Annots.0.3DD
(e)	Page1.Annots.0.3DD.OnInstantiate.ASCII stream
(f)	Page1.Annots.0.3DD.Binary stream
(g)	Page1.Annots.0.3DV
(h)	Page1.Annots.1

Table 1: *Levels of the COS-tree shown in Fig. 5.*

For each page there is one dictionary (a) containing the page dimensions as four floats in a MediaBox (shown in (b)), and the Type, which is Page. In "normal" PDF documents the page content such as text and images is contained in a Page.Contents node. In our case, however, the document consists only of a 3D model. Interestingly, this model is treated by PDF as an annotation; we call it the *3D model annotation*. This annotation, shown in (c), has the type Annot with sub-type 3D. The 3D model is not contained in the Contents (which state just 3D Model), but in a 3DD node. The 3DD.Binary Stream (f) contains the actual 3D model; PDF supports two 3D file formats, the U3D standard and the PRC format from Adobe.

The 3DD node has a very interesting structure (d) : 3DD.OnInstantiate contains *JavaScript code* that is executed when the user activates the 3D model in the PDF viewer. This code can register mouse callback functions to perform special actions. It has full access to the hierarchy of transformations in the 3D scene, so in principle very complex behaviours and animations are possible. In our case (see Fig. 6) we have chosen to only set a special camera view and to activate a hyperlink (lines 21,22) when the user double-clicks (lines 4,5) on one of the specially inserted models (line 9) of a ***clickable 3D annotation***. The JavaScript code that is inserted is the same for all our generated PDFs, except for the array of hyperlinks in line 1.

Besides 3DD, the 3D model annotation contains 3DV (g), which defines the standard view: 3DV.C2W contains the $12 = 3 \times 4$ float values of the camera transformation matrix. Other viewpoints can be added to 3DD.VA (d), which contains the standard views 0-5 (top, front etc.) and, in this case, three additional custom views 6,7,8, inserted by our software.

4.2. Trade-offs of different types of 3D annotations

Besides the mesh annotation there is additionally a text annotation, Page1.Annots.1 shown in (h). In this case it con-

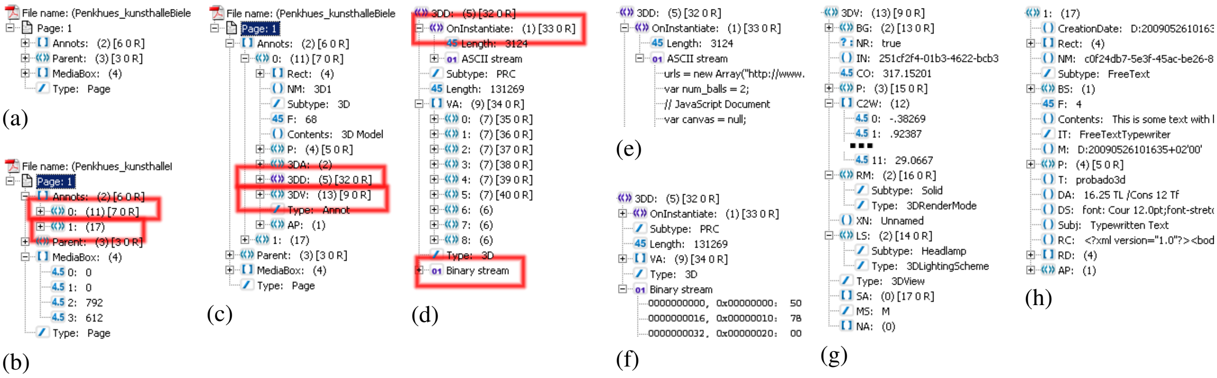

Figure 5: *First levels of the COS-tree, the internal structure of a PDF document. From left to right, successive levels are unfolded, as described in Table 1.*

```
1  urls = new Array("http://www.eg.org/","http://www.epoch.org/");
2  // JavaScript Document
3  var handler = new MouseEventHandler();
4  handler.onMouseDoubleClick = true;
5  handler.onEvent = function (mouseEvent) {
6    // Hits are depth sorted, so first mesh hit is clicked object
7    for (var i = 0; i < mouseEvent.hits.length; i++) {
8      var m = mouseEvent.hits[i].target;
9      if (m.constructor.name == "Mesh") {
10       // Find out the index of the hit mesh:
11       var index = −1, other = 0;
12       for (var i = 0; i < scene.meshes.count; i++) {
13         if (scene.meshes.getByIndex(i).constructor.name != "Mesh")
14           other++;
15         if (scene.meshes.getByIndex(i).name == m.name)
16           index = i − other;
17       }
18       if (index > 0) {
19         var camera = scene.cameras.getByIndex(0);
20         camera.target = scene.meshes.getByIndex(index + other);
21         host.getURL(urls[(index−1)], false);
22       }
23       return;
24     }
25   }
26 }
27 runtime.addEventHandler(handler);
```

Figure 6: *Example of JavaScript code to be executed whenever the 3D canvas is activated. It is contained in (e) from Fig. 5. Only line 1 is depends on the annotations.*

tains a simple string that is overlaid to the 3D canvas. All standard PDF annotations, which can be inserted by the *Comment & Markup tools* of Acrobat Professional, can in fact be used (e.g., sticky notes, stamps, etc). An example of such annotations is shown in Fig. 8, which demonstrates that they can even be embedded in 3D; these are called ***3D comment annotations***. The disadvantage of these "official" PDF annotations that are explicitly listed in the COS-tree, however, is that there are no JavaScript event handlers for them, e.g., to trigger complex behaviour or to attach hyperlinks. Still, they are an effective means of communication, since researchers can collaboratively comment on details of 3D models, and easily exchange them, e.g., via e-mail. Currently, 3D comment annotations can be comfortably inserted only using commercial software, e.g., *Acrobat Professional*. But since they are part of the PDF standard, the 3D comment annotations can also be created using other software, which is desirable for two reasons:

- 3D comments can be created in CH tools that are free
- 3D comments in PDF alone are not sustainable. In the common infrastructure scenario, the CH software should make sure that important annotations are saved in the PDF *and* in the common repository.

The flexibility to use JavaScript in clickable 3D annotations implies another serious drawback: The 3D model, i.e., the embedded 3DD.Binary stream, must be changed. To make the code in Fig. 6 work, appropriate meshes (named "Mesh"), e.g., to depict transparent balls, must be inserted into the stream for each annotation. Any extraction of the mesh from the PDF will yield also these annotating meshes; this clearly impairs the suitability of this approach for long-term archival. However, we have tried to make it as easy as possible to identify these annotating meshes, so there is hope that this will be possible also in the distant future.

It must be noted that we used the PRC format and not U3D in the examples. Despite the fact that U3D is clearly preferable because it is a standard that is independent from PDF or Adobe, the API support for PRC is much better than for U3D [Acr08]. This made the manipulation of the 3D stream, and thus our implementation, much easier. We will switch to using U3D as soon as we have our own tools in place to convert a given mesh to a U3D stream that contains also our annotating geometry. At this point, we will also clearly document all types of annotation objects that are used, in order to obtain long-term sustainability but still have the flexibility to use JavaScript for clickable 3D annotations.

4.3. EpochPDFpublisher and EpochPDFplugin

bns,Head, 3.89, 61.36, 9.15, 6.60, 3.89, 61.36, 9.15, http://www.eg.org/,Enter title,
bns,Hand, 6.97, 43.24, 9.26, 3.44, 6.97, 43.24, 9.26, http://www.epoch.org/,EPOCH,
0,view,MyView, -1,0,0, 0,0,-1, 0,1,0, -14,-150,-32
0,view,MyView2, -1,0,0, 0,0,-1, 0,1,0, -14,-150,-32
1,view,MyView, -1,0,0, 0,0,-1, 0,1,0, -14,-150,-32
0,text,418,51,757,99,This is some text with linebreaks\nthat we want on the page.

Figure 7: *Example of a CSV table used as input for the conversion. It contains two clickable 3D annotations (ball type), three additional view points, and an overlay text annotation*

The conversion is triggered by calling the EpochPDFpublisher, a stand-alone executable that takes as input a 3D file, e.g., in VRML format, and a table in simple CSV (*comma separated value*) format to describe the mesh annotations, viewpoints, and textual annotations to be added:

```
EpochPDFpublisher.exe --input myfile.wrl --csv myfile.csv
                      --output myfile.pdf
```

Using a CSV file means that the interface for the creation of annotated 3D PDF models is very robust, clear, and simple to use. A typical CSV file is shown in Fig. 7. Its first line describes a clickable 3D annotation, with type bns for transparent ball, (x,y,z) coordinates of the center, r radius, (x,y,z) text anchor point, URL to link to, and text to display in 3D. The URLs are to be passed on to line 1 of the JavaScript code. In the CSV, two ball annotations are followed by three viewpoints, which are to be inserted into the drop-down box of the 3D navigation pane in the viewer, to comfortably choose between views with smooth animations. The last line describes a text annotation that is to be inserted as "official", PDF-conforming, but non-clickable annotation into the Page.0.Annots.

The current version of EpochPDFpublisher relies on secretly launching a remote-controlled instance of *Acrobat Pro Extended*. It uses the Acrobat SDK to convert the input model to a single-page PDF document with a default structre. It closes and re-opens this document, this time triggering a special purpose plugin of the Acrobat Pro Extended, the EpochPDFplugin. The publisher sends the CSV data to the plugin, curiously using a hidden window and window messages as some sort of connection socket, which in fact is the recommended technique by Adobe. The plugin can make use of the full capabilities of Acrobat Pro Extended to make changes to the COS-tree to insert 3D comment annotations and viewpoints, and it gives access to functions to manipulate the PRC data stream containing the model. Finally it closes the modified PDF, which is now ready to be viewed by any 3D-enabled PDF reader. To summarize the workflow, this happens when the *"Publish PDF"* button is pressed:

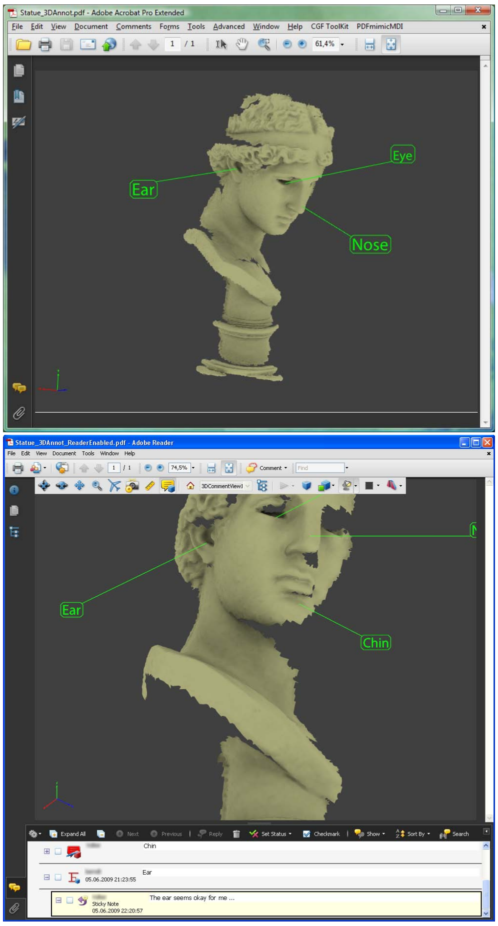

Figure 8: *Collaborative commenting of 3D content.* Top: *Researcher A adds 3D comments to objects and enables commenting for the Acrobat Reader.* Bottom: *Researcher B uses the Acrobat Reader to add new comments and can also reply to the comments of A.*

	EpochStudio	creates VRML and CSV
→	EpochPDFpublisher	launches AcrobatProExt
→	AcrobatProExtended	creates initial PDF
→	EpochPDFpublisher	sends CSV data to plugin
→	EpochPDFplugin	reloads and modifies PDF
→	EpochPDFpublisher	saves the created PDF

5. Results and Discussion

Manipulating 3D annotations in PDF is certainly not a straightforward practice. One needle's eye of the current implementation is the need for a controller application to contact an Acrobat plugin if anything inside a PDF 3D data stream needs to be modified. The SDK is quite cumbersome indeed, and errors in the mesh manipulation using the SDK tend to produce rather error prone 3D data streams that are difficult to "debug".

The annotated 3D PDFs we have produced nicely perform the task of archiving annotations. However, Acrobat Reader yields a rendering quality of the annotations that is not up to par with current technology. The balls we inserted into the 3D annotations, for example, show ugly artifacts.

The EpochStudio permits attaching multiple URLs to the same spot, because in the CH context these URLs are considered interpretations; and there will typically be more than one valid interpretation. We have not found a good solution for attaching multiple URLs to the same location using the JavaScripts included with the 3D model. It would for instance be convenient to open a dialog window that lets the user select one of the attached URLs. This is not possible with the JavaScript code in the 3D annotation, but there is a chance that it might be achievable using the JavaScript embedded on *document level*.

For highly detailed models the conversion to the PRC format can take a significant amount of time, e.g., two minutes for a PLY file of 20 MB.

To satisfy intellectual property protection requirements, and to reduce the file size, the 3D model can be embedded in simplified form. In addition Acrobat offers the option to either "allow" or "prevent" extracting the 3D data from the PDF document, which is not really secure because 3D data can always be read on the graphics driver level. The standard security features of PDF, such as or digital signatures, can also be applied to PDF documents containing 3D.

Collaborative review and knowledge exchange can be easily enabled by activating the option *"Enable For Commenting and Analysis in Adobe Reader"*. Then any user can add comments and save them using the gratis Acrobat Reader (see Fig.8).

6. Conclusion and Future Work

We have identified a whole range of different possible ways of using the 3D capabilities of PDF in Cultural Heritage beneficially. The ubiquitous availability of Adobe's PDF Reader and Acrobat provides for a nice, portable format to supply end-users with annotated 3D models, i.e., models containing not only shape but also interpretation. Embedded JavaScript code allows for great interactive illustrations that can be defined by any 3D content authoring software generating PDFs. We have exploited the JavaScript capabilities of 3D models for attaching information to locations on the surface of 3D models. Other uses would be camera-flights or animating 3D models to show alternative spatial arrangements of historical artefacts.

An area to explore further is interactive PDF (and 3D PDF) manipulation controlled by a webservice. PDF seems to be versatile at communicating with web servers, although this is apparently not possible using JavaScript on the level of 3D annotations. Document-level JavaScript has a so-called Net.HTTP object that might be usable for giving control over the PDF to a webservice.

The current solution makes use of the Acrobat SDK for manipulating the PDF document. The drawback is that this requires the commercial software *Adobe Acrobat Pro Extended* to be installed. This makes it impossible to use it in a web service solution due to license issues. An alternative would be to use the PDFLibrary SDK from Adobe, which provides the same API interface as the Acrobat SDK, but can be used to develop stand-alone application. Pricing is royalty based and depends on the Adobe PDF Library functionality used by the application. Another alternative is the commercial PDF3D SDK from Visual Technology Services, a C++ library that permits direct publication of the 3D content. This is very useful when adding PDF export capabilities to existing applications.

Acknowledgements

The authors would like to thank the Gipsmusuem of the Institute of Archeology, Graz University, in particular Dr. Manfred Lehner, for providing the statues used in this paper. This work was partially supported by the German Research Foundation DFG under grant INST 9055/1-1 (PROBADO project http://www.probado.de/). The authors gratefully acknowledge the generous support from the European Commission for the integrated project 3D-COFORM (www.3d-coform.eu), grant no. ICT 231809.

References

[Acr08] *Acrobat 3D API Reference, version 2.0.* Adobe Systems Incorporated, 2008.

[BF08] BARNES D. G., FLUKE C. J.: Incorporating interactive 3-dimensional graphics in astronomy research papers. *New Astronomy 13*, 8 (Nov. 2008), 599–605.

[CCC*08] CIGNONI P., CALLIERI M., CORSINI M., DELLEPIANE M., GANOVELLI F., RANZUGLIA G.: Meshlab: an open-source mesh processing tool. In *Sixth Eurographics Italian Chapter Conference* (2008), pp. 129–136.

[HSBE08] HAVEMANN S., SETTGAST V., BERNDT R., EIDE Ø.: The arrigo showcase reloaded – towards a sustainable link between 3d and semantics. *Proceedings of the 9th International Symposium onVirtual Reality, Archaeology and Cultural Heritage (VAST) 9* (2008), 125–132.

[ISO05] ISO 19005-1:2005: *Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1).* ISO, Geneva, Switzerland, 2005.

[ISO08] ISO 32000-1:2008: *Document management – Portable document format – Part 1: PDF 1.7.* ISO, Geneva, Switzerland, 2008.

[Kad05] KADOBAYASHI R.: Automatic 3d blogging to support the collaborative experience of 3d digital archives. In *ICADL* (2005), pp. 109–118.

[Vis09] VISUAL TECHNOLOGY SERVICES: PDF3D, 2009.