

The Presentation of Cultural Heritage Models in Epoch

Sven Havemann¹, Volker Settgest¹, Dieter Fellner^{1,2}, Geert Willems³, Luc Van Gool³, Gero Müller⁴, Marting Schneider⁴, Reinhard Klein⁴

¹Institut für ComputerGraphik & WissensVisualisierung, TU Graz, Austria ²TU Darmstadt & Fraunhofer IGD, Germany

³Center for Processing Speech and Images (ESAT-PSI/VISICS), K.U.Leuven, Belgium

⁴Universität Bonn, Institut für Informatik II, Germany

Abstract

The presentation of CH artefacts is technically demanding because it has to meet a variety of requirements: A plethora of file formats, compatibility with numerous application scenarios from powerwall to web-browser, sustainability and long-term availability, extensibility with respect to digital model representations, and last but not least a good usability. Instead of a monolithic application we propose a viewer architecture that builds upon a module concept and a scripting language. This permits to design with reasonable effort non-trivial interaction components for exploration and inspection of individual models as well as of complex 3D-scenes. Furthermore some specific CH-models will be discussed in more detail.

1. Introduction

Cultural objects in museum exhibitions are sometimes not easy to appreciate. They are too small, very detailed, behind glass, and too precious and delicate to let them be touched by visitors. It is particularly difficult to study and appreciate the amazing, intricate detail and the traces a long history has left on cultural artefacts.

3D technology can help tremendously to enhance the appreciation of museum objects. Our guiding vision is the idea of a *complementary exhibition*: Real artefacts in a museum exhibition are complemented by digital artefacts whose sole purpose is to deepen the understanding and appreciation of the real ones. A particular form of a digital artefact is the *digital replica*. To show it in a museum combines the authenticity of the real with the ease of manipulation of the digital. As a result, museum visitors become more engaged since they can actively participate. This approach is also quite successfully used in science and technology museums, whose number has greatly increased over the last years. School children are encouraged to actively acquire knowledge by trying out scientific experiments. – Cultural museums are unfortunately still lacking behind in exploiting this *edutainment* aspect.

It is important to emphasize that we do not recommend the use of technology under all circumstances. We advocate instead designing *modest systems*, where never the technology is in focus, but always the content. This implies, for instance, that technological gadgets must be extremely easy to use. They shall not require manuals or instructions, they must work robustly and, most importantly, they must behave in a predictable way. No bad surprises, no frustration, because that takes away visitors' attention from the artefact.

The same, however, applies to the CH professionals who preserve and present our common historical knowledge: We must offer them *enabling technology* that adapts to their needs and procedures, and not impose or presuppose a technologically biased view.

1.1. Strategic Vision

It is essential for Epoch that all the beautiful ideas and approaches as, e.g., shown on the VAST series of conference, find their way to the public audience through museums and exhibitions. In terms of interactive 3D presentations, much more is possible than what can be found today in average museums. We want to change this situation and make using interactive 3D a standard. The key features of our solution are therefore usability and affordability. The technical agenda of the Epoch project is based on the idea of a CH pipeline, a complete workflow that ranges from the archeologist that finds an artefact in the field to the presentation of the artefact in a museum. We think we should start to create a demand for content at the end of the pipeline in order to stimulate the development of a CH market based on Epoch technology on the whole pipeline.

1.2. A Concrete Target Scenario

A museum curator decides to make a new exhibition on the Etruscan period. He hires a company that digitizes the tiny golden brooches and small pieces of precious jewelry he is going to present. From a recent excavation campaign he purchases the scanned 3D datasets of different strata of the archeological site where the beautiful historic artefacts were found, as well as laser scans of the remains of houses, pillars, statues, etc. He puts all the digital artefacts into the 3D presentation software, chooses a theme (skin) that fits with the look of the physical exhibition, and uploads the new presentation to the 3D multimedia kiosks via network.

In the exhibition, visitors can see all the real artefacts, as usual behind glass. But in every exhibition hall there is also a 3D multimedia kiosk where the visitor can inspect the small beautiful artefacts interactively from all sides, which is not possible with the real. The visitor can also have a look at the archeological site where the artefact was found, which also brings up a few statues and other findings.

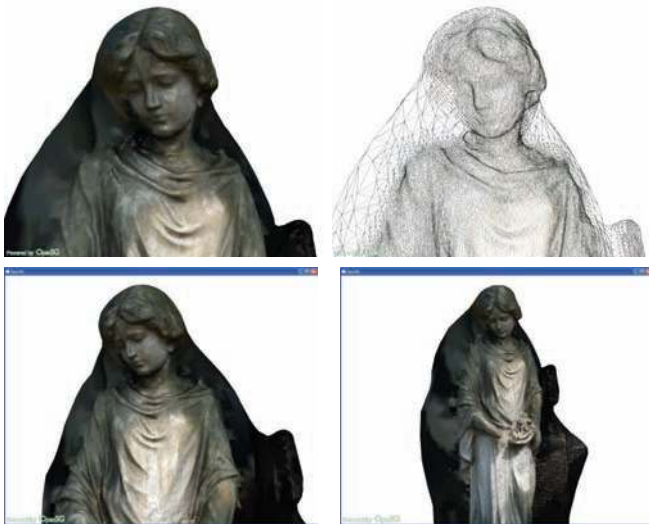


Figure 1: Interactive inspection of a textured triangle mesh. The wireframe reveals how sparse the dataset actually is. A single (r, g, b) -texel per surface point gives realistic results only for very matte (Lambertian) surfaces, but not at all for complex, shiny materials.

1.3. The duality of Acquisition and Presentation

The purpose of this paper is to present some of the technical infrastructure that is necessary to let the described vision become reality. Creating an infrastructure is the focus of this paper, and also a great challenge: So far only a vast number of specific solutions to specific problems exist, usually described as research prototypes in scientific papers. It is very difficult to identify a common classification that applies to all of them. We follow a very rough schema using two classes to structure this paper:

- **Acquisition:** Produce sustainable digital replicaes by measuring
- **Presentation:** Interactive exploration of a the acquired datasets

To support a specific shape representation means for the common infrastructure that solutions are operational for three things: for acquisition, for long-term storage, and for presentation.

The shape representation problem. The link between acquisition and presentation are obviously the datasets. The difficulty in designing a common infrastructure comes from the fact that there is a myriad of techniques to create and to store these data. Literally thousands of slightly different encodings (*file formats*) exist. But which of them to choose to support in the common infrastructure? Each has its strengths and weaknesses, each of them is being used within its particular community. For most of these communities, a rock-solid, sustainable long-term compatibility is not an issue, since this implies a development overhead that is not tolerable. But for CH as such, however, sustainability is crucial: Recorded historical evidence is supposed to be available forever.

The problem is illustrated in Figs. 1 and 2, which also allude to the complexity and the subtlety involved: Textured triangle meshes are probably the most-used data type for CH documentation today. On the one hand they are much too simplistic: Fig. 1 shows the incredible loss of information from the twelve 6 MP images (input data) to the sparsely sampled 3D mesh with vertex colors, i.e., only

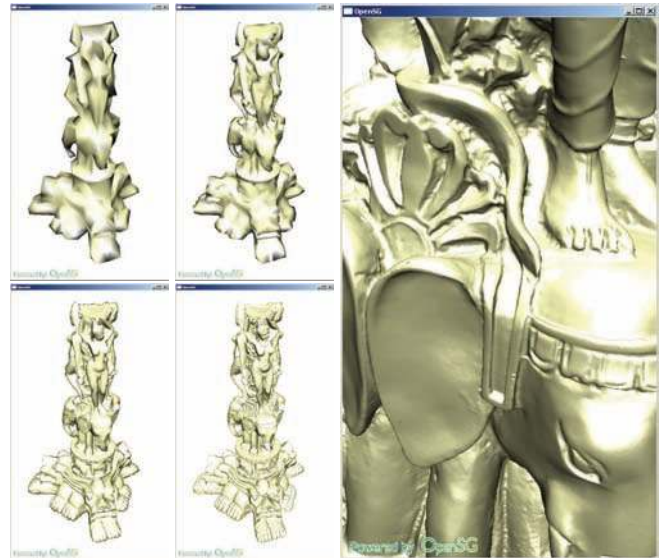


Figure 2: An un-textured Nexus object. Nexus is a file format for massive multi-resolution meshes developed at CNR Pisa. It allows to first load very quickly only a rough surface approximation, which is then progressively refined during the interactive exploration to reveal finer surface details.

a single (r, g, b) triplet per mesh vertex (output). On the other hand triangle meshes are too complex: The size of the dataset in Fig. 2 is 526 MB, which is much too much keeping in mind that a typical display has only 2 million pixels. The obvious answer is to use multi-resolution meshes; but for this particular sub-problem alone hundreds of solutions exist, from Hoppe's Progressive Meshes to Cignoni's Block-maps.

1.4. Overview of the paper

The *Epoch common infrastructure* supports following shape representations with solutions for acquisition, storage, and presentation:

- **Synthetic reconstructions** created with standard 3D packages. The file formats of Max, Maya etc are for sustainability reasons not supported directly but only exported meshes (obj,ply)
- **Scanned 3D models** acquired by laser scanning or structured light are processed, e.g., using Meshlab to produce static (obj,ply) or multi-resolution models (Nexus, see Fig. 2)
- **Photogrammetric reconstructions** are obtained from sequences of photos that are converted to range maps by the Epoch Webservice [VG06] and then merged using Meshlab
- **Procedural cities** are synthetically generated using a shape grammar that encodes architectural rules [MVW*06] optionally taking into account archeological GIS data of building ground layouts.
- **Parametric shapes** are generated from a few high-level parameters by generative mesh modeling using the [Generative Modeling Language](#) [HF04, Hav05].
- **Scene graphs** contain a hierarchy (a tree) of nodes that are either coordinate transformations or individual objects (of the aforementioned types). Epoch uses the [OpenSG](#) scene graph, scenes can either be read from *Collada light* files [HSKF06] or from .osb files used for non-archival purposes.

New acquisition devices can have fundamental implications for also for the presentation, namely when new channels of information become available. Accordingly, the rest of the paper has two main concerns:

Acquisition: Models created using camera domes. This technology has a huge potential for the high-quality acquisition of CH artefacts with complex material on a mass-scale. However, including the produced models into the common infrastructure is also a technical challenge. To illustrate this, two recently developed acquisition domes will be discussed that both allow for the acquisition of geometry and appearance, but in slightly different ways.

Section 2 presents a single-camera dome for capturing the color and local surface orientation of objects by taking pictures where only the illumination direction varies. This allows not only for interactive relighting of CH models but also yields high-resolution surface mesh. This dome is also transportable to make on-site acquisition feasible.

Section 3 presents a multi-camera dome where the viewing and lighting directions can be varied independently. The captured data allow computing for each surface point a BTF, which is a much richer approximation to the true reflectance characteristic than standard texturing.

Presentation: The software infrastructure for the Epoch viewer.

The Epoch viewer is not a stand-alone application, as the name might suggest, but rather a versatile toolkit that is supposed to be useful for all sorts of CH-related visualisation tasks. The underlying software infrastructure grew out of the 3D-Powerpoint tool developed in the 3D-KIOSK sub-project of Epoch (a so-called *Newton*).

Technically the Epoch viewer is a software component (under Windows an ActiveX control) showing only a 3D drawing canvas for interactive display of the OpenGL scene graph. This modest design makes it very flexible to use, since it can be embedded into a custom-tailored conventional GUI to give it the familiar appearance of a normal application. This way it can be used in a variety of ways, from scholarly studies (browsing of digital libraries) to public presentation in museums. Note that the latter has some features in common with Microsoft Powerpoint, which justifies the name *3D Powerpoint*:

- It has an authoring and a presentation mode.
- It is based on customizable presentation templates.
- It is made for non-programmers focusing on content.

This infrastructure is presented in more detail in sections 4 and 5.

2. The Camera Dome in Leuven, Belgium

Our acquisition system, first proposed in [WVM*05], allows for easy and cost-effective digitizing of small CH artefacts. The benefits of our system are the following:

- the hardware contains no moving parts and can be easily (dis)assembled, which makes it ideal for transportation and on-site acquisition.
- the many lightsources, combined with a robust photometric stereo algorithm, allow for the automatic and accurate extraction of the object's color, local surface orientation and depth map.
- the digital artefact can be studied by virtually relighting it, while several shaders are available to aid the scholar with the transcription of digital cuneiform tablets.

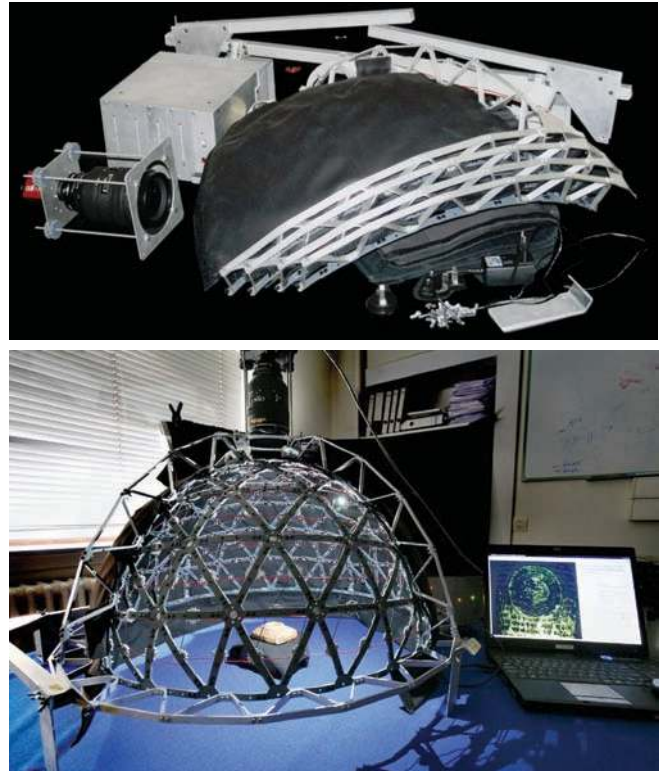


Figure 3: The acquisition hardware. (top) Overview of all the separate components ready to put into a suitcase. (bottom) Typical acquisition setup.

2.1. Motivation: CH documentation

Vast digitizing projects can be found throughout the last decade, and digital libraries are becoming increasingly mainstream. Besides the obvious reasons to digitally store and visualize artefacts, more and more digital models become the object of study for scholars as a surrogate for the actual artefact. The field of ancient Mesopotamian studies is a case in point.

The customary method to register and study cuneiform texts is a slow and often subjective process. The study requires a scholar with a specialization covering the specific contents, place of origin and time period of the text, and a scientific effort which can only take place at the location where the written document is kept because of conservation and political reasons. Cuneiform collections are spread all over the world, an additional disadvantage forcing scholars to travel around, which is both expensive and time-consuming. Identification or transcription of the cuneiform signs is best achieved by taking the cuneiform tablet in one's hands to look for the best lighting which makes the signs legible.

To allow verification by future scholars, the cuneiform signs are interpreted by a specialist and an accurate line drawing of the text is produced. This is not only time-consuming, but correct transliteration and translation is thus completely dependant on the competence of the sole specialist who studied the specific text at a specific location.

2.2. Related Work

The *Cuneiform Digital Library Initiative*, a joint project between universities and museums, is one of the best known digitizing projects for cuneiform tablets and uses flatbed scanners to make 2D images and hand copies available on the internet. As a means of preservation, however, 2D images have inherent shortcomings as much of the three dimensional aspects of the clay tablets is not captured. Over the years, many solutions have been proposed to also capture 3D information by using laser and structured-light scanners.

Most recently, Hahn et al. [HBD07] have presented a hardware solution within the *iClay/Digital Hammurabi* project, which combines structured light with photometric stereo from a sparse set of images, using a DLP projector, a turn-table and a CMOS camera, to obtain a 2D+ and 3D model of tablets. Their set-up however requires moving parts and may therefore be unsuitable in some out-of-lab situations.

Mudge et al. [MVSL05] opted for a 2D+ representation of tablet using *Polynomial Texture Maps* (PTM) [MGW01] which approximate the luminance information at each pixel with a biquadratic polynomial. When a denser shape description was desired, the PTM data was registered with range data, again obtained via structured-light solutions.

Malzbender et al. [MWGA06], on the other hand, proposed a purely photometric stereo solution that recovers per-pixel estimates of surface orientation. While their solution has the great advantage of working in real-time, it does not account for specularities and shadows and is therefore not robust enough to extract high-quality shape information.

2.3. The Acquisition Process

The acquisition hardware consists of a upper-hemisphere of about 35 cm diameter on which 260 LEDs are uniformly spread out. A camera is positioned on the top looking down at the center of the hemisphere. The lights are being controlled via a custom built USB controller, while the camera is hooked up to the laptop via Firewire.

Furthermore, the dome was designed with portability and easy-of-use in mind. The hemisphere can be easily disassembled into 4 parts which, together with the supporting structure and USB controller, fit into a normal suitcase. The combined weight does not exceed 10 kg and build-up and tear-down take a mere 15 minutes. Combined with the camera and a powerful laptop, the whole system is therefore ideal to be used on-site. In figure 3, an overview of the disassembled components can be seen (top), together with a typical acquisition setup (bottom), ready for acquisition.

2.4. Processing Pipeline

Photometric stereo allows the estimation of local surface orientation and albedo by using several images of the same surface taken from the same viewpoint but under illumination coming from different directions. In order for a single side of a artefact to be digitized, it is therefore sufficient to position it in the center of the dome. After adjusting the lens and camera settings, a push of a button start the acquisition process which takes a sequence of images, each time with another LED being lit.

While standard photometric stereo algorithms assume that the observed surface has a Lambertian behaviour, this is rarely the case. In reality, many artefacts have a reflective surface (e.g. coins) or a

surface with a lot of depth variation (e.g. cuneiforms with sealings) that result in self-shadowing. Both cases render the Lambertian assumption invalid and force us to resort to a method which is robust against such complicating effects. An iterative approach has been proposed which computes the photometric stereo while selecting for each pixel those images for which the Lambertian assumption holds.

As we compute a normal at each pixel, a high-quality 3D surface can be obtained by calculating the depth map via multigrid-based normal map integration, as shown in Fig. 5. For more in-depth information, we refer the interested reader to [WVM*05, VWV06].

2.5. Digital Tablets

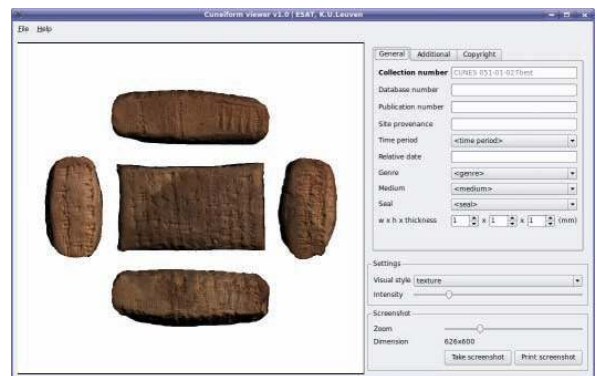


Figure 4: Viewer application allowing for real-time rendering of virtual cuneiform tablets using many different shaders. Five sides of the unfolded tablet are shown in the left part of the split layout while the meta-data together with shader and print options are shown to the right.

The virtual counterpart of a cuneiform tablet consists of a file containing all relevant meta-data together with a binary blob for each digitized side, which contains an albedo and normal map. As typically a high-resolution camera is used for acquisition, some compression is advised to keep the overall file size practical. The albedo maps are stored with 3 bytes per pixel using standard image compression techniques. The normal, on the other hand, are quantized and compressed from 3 floating point values into 2 bytes. Furthermore, all maps are cropped to their relevant regions, as selected by the user during acquisition. This results in a typical file size of 40MB for a virtual cuneiform tablet where all 6 sides of a tablet captured with a 5MP camera.

Once a cuneiform tablet has been digitized, the data can be loaded into a specialized viewer application, shown in figure 4, which allows the user to interact with the tablet in a similar way as if he was holding the actual artefact. Alternatively, a 3D polygonal model can be created from the extracted depth map. The availability of a tablet becomes therefore as easy as sending a file over the internet.

The tablet is typically shown as a unfolded cube, allowing the scholar to read 'across edges'. The viewer allows the user to interactively shine a virtual light onto the virtual tablet with variable direction and intensity. By implementing the visualization purely on the GPU, all renderings are computed in real-time using the full resolution of the computed maps. Furthermore, several filters can be applied to enhance specific regions. Several of these filters are shown in figure 6. To the left, a typical relighting is shown which

simulates a raking light over a tablet. The shaded version (middle) is obtained by showing the tablet without its color, which can improve the reading in some cases. At the right, the result from the line-drawing filter is shown which tries to separate the wedges from the tablet. This output can be used as a basis for a more objective hand copy as discussed in section 2.6.



Figure 5: 3D reconstruction of a tablet with prominent sealing. (left) obtained dense normal map from robust photometric stereo, (right) reconstructed 3D model.

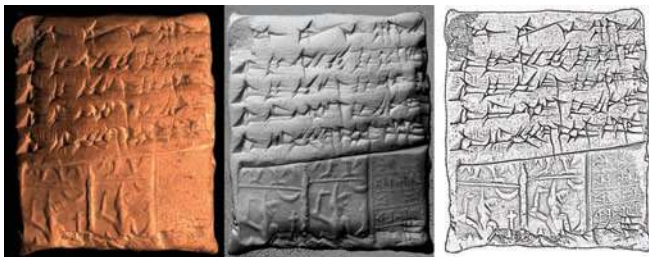


Figure 6: Three different visualizations of the reverse of tablet CUNES 51-01-015. From left to right: virtual relighting without specularities, shaded version without albedo and extracted line drawing.

2.6. Dissemination

The first trials of the hardware and software were run on the cuneiform collection of the Katholieke Universiteit Leuven. The benefit of this approach was that these texts were already published well in the past (OLP 3), the new recordings could therefore be evaluated alongside the result of this previous study. The recording, processing and application of the software were executed by the scholars of the Near Eastern Studies unit. Based on the feedback of these experiences, updates could be incorporated in the system. A second test phase was held at the Cornell University (NY, USA) on a never hitherto studied section of a voluminous collection. The Cornell collection reflects the typical complexity of a body of texts dating to various periods, coming from divergent provenances and encompassing different text genres. In this context, scholars of the Near Eastern Studies unit were asked to study those texts in the collection CUNES (see figure 6) dating to the late Old Babylonian period (ca. 1700-1600 BCE). To record the tablets, the hardware was transported in a standard suitcase and during a month all aspects of the system were thoroughly tested. The method of recording proved to be quite user-friendly for non technically educated researchers. Back in Belgium the texts could be studied successfully using the recorded results and by applying the digital filters in the viewer software.



Figure 7: The acquisition, reconstruction, and rendering of a CH artefact using the camera dome from Bonn. The workflow is explained in Fig. 9.

The implementation of the system, especially on the Cornell University Cuneiform Collection, made several important benefits come to light. Instead of a long stay at the institution where the tablets are kept, only a short stay was now needed. The relighting application in the viewer did not only allow a clear identification of the signs, it also offers extremely good visibility of the sealings on the surface of the tablets. The study of these impressed sealings has always been difficult, but the combined use of the relighting system and the digital filters renders them clearly visible. One of the filters automatically extracts line drawings, which proved to be most useful for the correct interpretation of cuneiform signs and the sealings. Furthermore, this allows for a time-efficient and objective basis for the transliteration and translation process. To enhance the contrast between the surface and the impressed signs, one of the available filters extracts all color, which substantially improves the readability of the text. Before, this same effect could only be achieved by covering the surface of the tablet with a thin white coat of ammonium chloride (NH_4Cl), an intensive treatment now abolished.

The overall CH digitizing system has been mentioned in local Belgian written press and television and is featured in Euronews' Futuris episode *European Digital Library to contain all knowledge* [epi07]. From december 2007 to april 2008, the system is also being displayed in the national exhibition "From Gilgamesj to Zenobia" at the Royal Musea for Art and History (KMKKG), Brussels, Belgium.

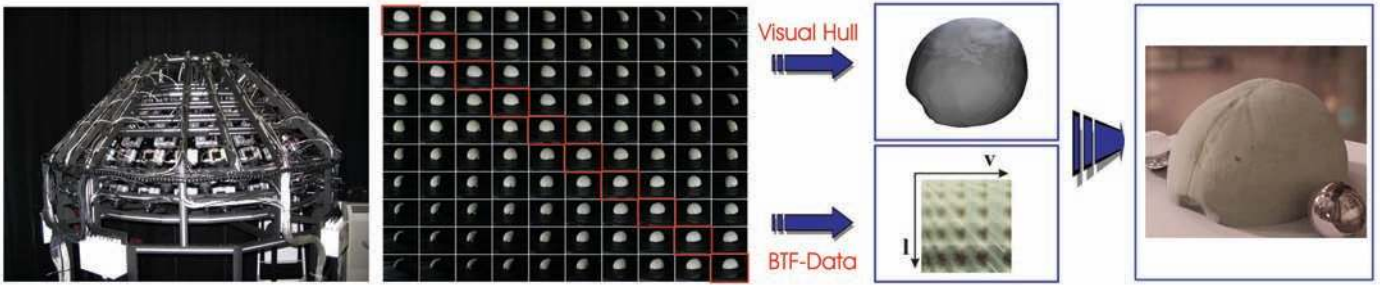


Figure 8: The Acquisition Pipeline: The multi-camera array records 151 images per camera per light direction, resulting in 22801 images. Here, only the first ten cameras (from left to right) and the first ten light directions (from top to bottom) are shown. From the full set of these images, the BTF is constructed, while only a subset (the diagonal in this matrix notation) is used for the geometry reconstruction.

3. The Camera Dome in Bonn, Germany

In order to produce visually appealing digital models of cultural heritage artefacts, a reconstruction of the 3D geometry alone is often not sufficient because accurate colour and reflectance information give essential clues for the object’s material. Standard texturing methods implies severe limitations of material and lighting conditions. This in turn limits very much the usefulness of a digital replica, because moving, rotating, and seeing an artefact in different lighting conditions (environment maps) are indispensable for a decent understanding.

3.1. Camera Dome Approach in Bonn

With this situation in mind, a novel high fidelity acquisition system (see Fig. 8) to synchronously capture an object’s 3D geometry and material properties in a very time-efficient and user-friendly way was developed in Bonn [MBK05, Ben07]. Our system exploits images from a multi-camera array to reconstruct an artefact’s coarse to medium scale geometry using a GPU-based visual hull technique, resulting in a closed triangle mesh. In parallel, the images are also used to capture the object’s appearance into so-called bidirectional texture functions (BTF) – a 6-dimensional texture representation introduced by Dana et al. [DNvGK97] which extends the common textures by dependence on light- and view-direction, and thereby allows for photo-realistic rendering of an objects micro- and mesostructure. The key contributions of our system are that it

- fully automatically acquires 3D-data, capturing an object’s geometry and its visual appearance in form of bidirectional textures,
- faithfully reconstructs the object’s mesostructure using BTF-techniques and therefore effectively overcomes the limited accuracy of the visual hull technique, and
- it is time efficient and very easy to use.

3.2. The Multi-Camera Grid

For fast acquisition of the images required to measure the BTF and to reconstruct the geometry of the object to be digitised, we use an array of 151 commodity digital still cameras mounted on a hemispherical gantry. By arranging the cameras into this array, the acquisition of the images required to construct the BTF textures is parallelised and no moving parts (e.g. a rotating stage) are needed. Therefore, the positions of the image sensors and the light sources can be calibrated

in a preprocessing step which only has to be carried out if a camera has been replaced or after the whole setup has been transported. The low-level post-processing (geometric correction, colour correction) is fast enough to be done in parallel to the measurement.

3.3. Geometry Acquisition

In our current implementation, we first extract the objects silhouettes from the acquired 2D photographs by simple thresholding. As we use no backdrop in our setup, we set every pixel with a brightness of less than a certain threshold to be *outside*, the remaining pixel are *inside*. Together with the viewpoint information known from our setup, every outside-pixel in each image now defines a ray in scene space that is known *not* to intersect the object, whereas the inside-pixel define rays that intersect the surface at some unknown distance to the viewpoint. In the continuous case (pixel width $\rightarrow 0$) the union of all these intersecting rays would define a generalised cone that is guaranteed to contain the object. As this fact holds for all acquired images, the intersection of all these generalised cones (the *Visual Hull*, [Lau94, MBR*00]) describes a tight volume in space in which the complete object must lie.

The large number of acquired images and the (potential) need for finer grids make it impractical to actually traverse the grid following the *outside*-rays. Instead, we use a hardware-supported approach where we interpret the grid as a stack of binary 2D textures. The *inside/outside*-information is then efficiently collected by projecting every source image to each texture in the stack using projective texture mapping.

3.4. BTF Acquisition

The image-based approach to capturing the appearance of an object for later rendering is to take dense sets of images under controlled viewing and lighting conditions in order to sample its reflectance field appropriately. Since fidelity BTF measurements have huge memory requirements efficient compression techniques are needed. We use a clustered PCA compression [MMK04] [MSK06] which allows for an easy decompression and rendering using graphics hardware.

Statistical analysis like PCA, however, requires that data entries in the BTF are semantically correspondent – an assumption that holds for the raw data only under the assumptions of planarity, orthographic projection and directional light sources. This is not the case

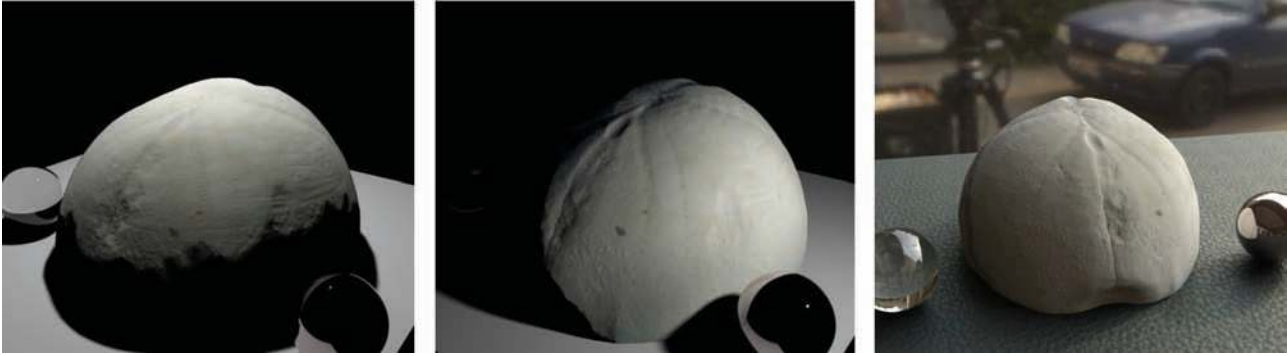


Figure 9: Raytraced renderings of a captured and reconstructed echinite under novel lighting and viewing conditions. The left and middle image are rendered with a small area light source and demonstrate the fine geometric details captured in the BTF. The right image shows a relighting of the echinite with a complex image-based lighting environment captured in front of the Pädagogische Fakultät in Bonn. The ground floor in the right image is covered with a synthetic leather BTF courtesy of DaimlerChrysler AG.

for our system since the dimensions of our acquisition setup cannot be considered "large" compared to the dimensions of the objects to be digitised. Therefore, we perform resampling of the raw BTF data based on a planar parameterisation of the reconstructed triangle mesh computed with the method of Degener et al. [DMK03] before compression.

Measuring BTFs generally consists of recording for every point on the surface its reflectance from each view direction under each light direction. For non-flat surfaces, however, the reflectance for some light and viewing directions will be zero (or close to) simply because of occlusion and/or self-shadowing. Using standard approaches, this missing information would be misinterpreted as a property of the material. Therefore, we first identify from all the points on the object surface those points that have incomplete measurements and perform a clustered PCA for them. The missing values of the remaining points are then completed in a way that the overall reconstruction error of them is minimized.

3.5. Results

Our approach to reconstruct the geometry from the acquired images using visual hulls computed on the GPU is reliable and fast. Of course, identifying a nonconvex object using a silhouette-based approach inherently and inevitably implies neglecting some features of its surface geometry. Despite this general seemingly inaptness of the visual hull reconstruction, we were able to produce realistic images of our captured objects (see Fig. 9) because the neglected surface features are well-captured in their appearance using the BTF texturing techniques. To further improve the geometric reconstruction of the object, methods presented in [HS03, CGS06] can naturally be incorporated into our system.

4. Presentation: The Epoch Viewer

The presentation of 3D objects to a public audience is often considered a solved problem since numerous possible approaches exist: 3D standard tools, professional 3D presentation software, game engines, scene graph engines, and certain academic approaches. Usually the first idea would be to use 3D standard tools such as a VRML/X3D viewer, 3D embedded in pdf (Acrobat3D), Shockwave/Flash-3D, etc. Also for 3D modeling tools such as Maya, 3DStudio, Blender



Figure 10: Models from the Herz-Jesu Church in Graz each reconstructed from 6-12 photos (6 MP) using the Epoch Webservice (KU Leuven) and the Meshlab software (CNR Pisa). Bottom: Water-tight models from the Poisson reconstruction filter in Meshlab, result of ambient occlusion shader, then multiplied with the acquired object color (Homer); and this with specular GLSL shader with per vertex color and per-pixel-lighting under OpenSG (Monster).

presentation plugins exist. But these "closed" solutions can be immediately ruled out since we target location-based presentations with high-quality CH content, and smooth interaction with non-standard input devices. We could certainly program extensions to, say, a standard X3D viewer, but then we would be tied to this particular piece of software over which we have no control.



Figure 11: Advanced interaction using GML scene graph scripting. Top row: Dragging the ball upwards unrolls the spiral staircase. Bottom row: The scene graph allows for very easy object distribution and orientation, for instance by using a coordinate frame computed from the surface normal at the picked position.

A much better option would be to use professional 3D presentation software such as Virtools, Quest3D, Shark3D, OfficeFX, and others. These tools provide professional rendering quality, support a wide range of input formats, hardware setups, and all possible input devices, and they have impressive feature lists, ranging from physics engines over audio to networking. However, an in-depth evaluation some years ago of a similar tool, Realimation, revealed some fundamental problems with such packages. They are

- **monolithic:**
Not a component, but a complete stand-alone application
- **proprietary:**
Vital features may change from one version to the next
- **not low-level extensible:**
They impose strict limits on what developers can access
- **not a modeler:**
Every non-trivial piece of geometry must be imported

It must be noted, however, that these tools provide a remarkable degree of usability: their authoring environments are extremely interesting, also from an academic point of view.

For most of the professional game engines – the 3D engines database on www.devmaster.net/engines currently lists 279 of them – basically the same considerations apply: Game engines such as Torque, 3DGameStudio, Ogre, or Irrlicht are optimized for efficiency and use the latest graphics hardware effects. This matches also the expectations of museum visitors, as more and more people are becoming acquainted with game technology. The downside, however, is that content creation for games requires much low-level realtime know-how, much programming, has a low long-time sustainability, and for serious non-game applications requires extra effort to get rid of the game overhead.

A much more acceptable alternative is to use a “neutral” scene graph engine such as Coin/OpenInventor, OpenScenegraph, OpenSG and the like. They combine efficiency and support of (almost) latest effects with openness and extensibility. In fact, they are

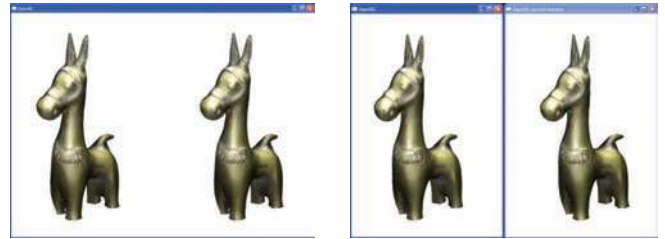


Figure 12: Stereoscopic display modes, and BTF materials. In *horizontal span* mode the desktop goes over two physical screens, whereas with *dual view* a fullscreen application can cover at most one single screen. When both channels are overlaid using a pair of projectors and a decent per-eye channel separation, a stereoscopic effect can be achieved. – The object shown has a BTF material that contains more than one texel per point. It allows to approximate the reflectance function (BRDF) at each surface point much better. The BTF was measured using the camera dome from Bonn (cf. Fig. 7).

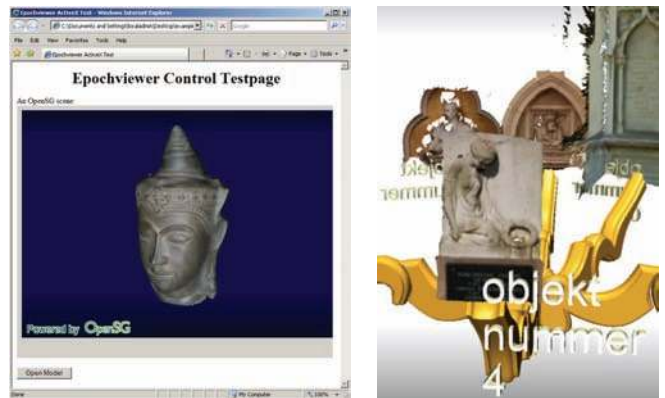


Figure 13: A whole range of display scenarios, from quick browsing of a model database using a web browser to stereoscopic high-end rendering in a kiosk application, are possible using the Epoch viewer with its module concept.

usually *designed for extensibility*. This is a huge advantage if, like in our case, we have to custom-tailor a 3D solution to the demands and existing standards of a particular application area, in our case cultural heritage.

There are only two academic approaches for 3D presentation environment for CH we have found. First, Costagliola et al. [CMFP02] publish a configurable presentation environment particularly for guided tours. The other work is the *Virtual Inspector* tool from our colleagues from Pisa, Italy [CPCS08, CPCS06], which focuses on the display of huge meshes.

Our approach differs from those in that our tool provides an event system that allows scripting of all sorts of interactive behaviours, operating on a scene graph that is assembled at runtime. OpenSG, the underlying open source scene graph is extensible, so that also the rendering of huge meshes is possible with a suitable render node.

However, through this research we have come to the following catalogue of criteria for our envisaged solution.

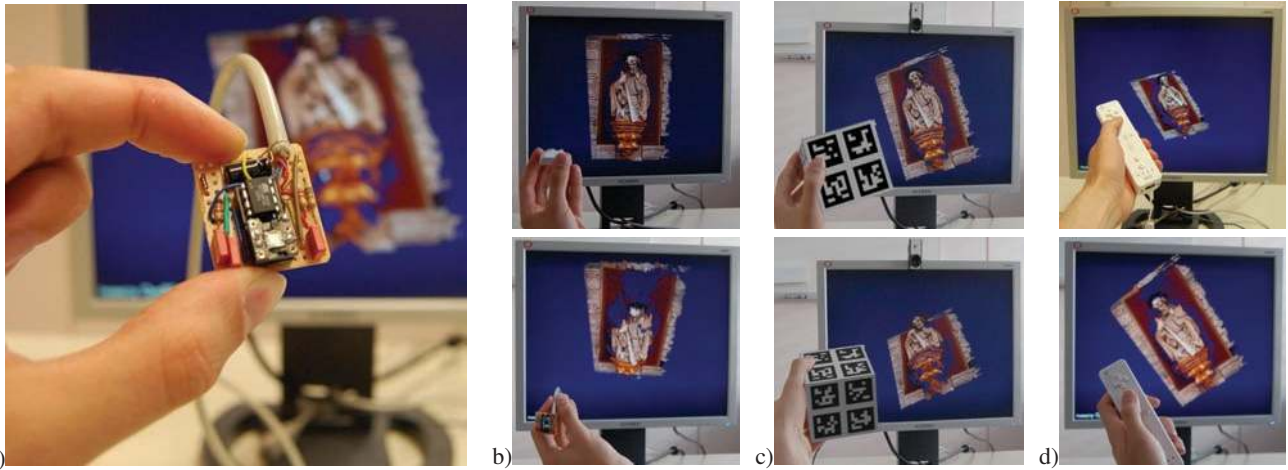


Figure 14: Tangible input devices: (a,b) Accelerometer, can measure 2 of the 6 DOF of a static pose, as it can detect in which direction gravity points. (c) Array of 2×2 ARToolkit markers, to make the camera-based detection more robust against occlusion by fingers. (d) Nintendo Wii controller, with accelerometer and integrated low-resolution camera, can in principle determine a 6-DOF pose plus 2-D acceleration.

5. Feature Wish List

Drag-and-Drop Authoring: Every presentation consists of *layout* and *content*. Both are easier to generate and arrange in 2D than in 3D. As we want non-3D-experts to design digital exhibitions, the best solution would be to have *pre-defined layouts* that can be *filled with content* from the local file system or from the internet via drag-and-drop.

Versatile Hardware Setups: A viewer should support a wide range of input and output devices (see Fig. 14). To support basically any human input device is possible using the concept of *channels*; using a well-defined communication protocol and socket communication, input devices can connect via network. The output options range from a 3D window in a web browser (see Fig. 13) to cluster rendering in a CAVE. Also stereoscopy is a desirable feature, but many flavors exist (see Fig. 12).

Easy 3D navigation: Once the presentation is defined it shall run in a public museum. Average museum visitors shall be able to use the 3D kiosk systems without having to read instructions. This is demanding in particular for the notorious problem of 3D navigation: Users shall never get *lost in 3D*, not reach impossible view points or see nothing on the screen, nor get locked somewhere. We want to allow as much 3D control as possible and restrict it only as much as is necessary to enforce consistency.

CH Integration: 3D presentation is only the very end of a long processing chain. Interoperability requires standards. The presentation environment should permit to make use of any additional information attached to cultural objects present, e.g., in the Collada 3D format used in Epoch. In the long run, even using CIDOC/CRM should be an option, as pointed out by Havemann et al. in [HSKF06]: Every cultural artefact is part of a network of semantic information. The ultimate form of a *Cultural Presentation Browser* would be a tool that allows average museum visitors to navigate through this semantic network.

CH Sustainability: 3D presentations will be a new form of collecting knowledge about history and culture. Ideally, our knowledge

about CH should be as long-lasting as the artefacts we show. This issue actually causes fierce reservations against the use of digital technology in the CH community. However, there is a tradeoff between using the latest technology and being independent from particular software and hardware versions. The solution we envisage is to use advanced, but *well-documented* file formats and algorithms. This way presentations can use state of the art hardware shaders and mesh compression, but are not deemed to be obsolete in five years.

Low-Level extensible: Today the usual form of representing scanned artefacts is as a textured triangle mesh. A serious 3D infrastructure, however, requires a more diverse set of shape representations. Scanned cuneiform tablets, for instance, usually have a multitude of view-dependent textures or under different lighting conditions, whereas computer tomography produces a volumetric “image” of what is inside, e.g., an Egyptian mummy [BSly]. This requires that new shape representations can be integrated with the viewer, e.g., loaders for new file formats, and new 3D rendering modules, for example for the BTF models from Bonn (see Fig. 12).

Cuneiform viewer module: The cuneiform tablets (see Fig. 6,2.5) so far require a viewer that is inherently 2D (shown in Fig. 4), and not 3D. It would be highly desirable, however, to overcome this artificial separation: The 2D relighting application should actually be embedded in 3D (as another low-level module), because only then the 2D view can be shown together with the 3D model. The way to go would be to generalize the 2D application in the sense that the 2D content is rendered on a rectangle floating in 3D that is part of the normal scene graph. – The only remaining issue is that advanced 2D viewers today also heavily use the GPU. A solution might be upcoming GPU frameworks like CUDA from Nvidia, which have mechanisms for management of the GPU resource.

Template extensible: Whereas the main use of the 3D kiosk is to let a visitor explore one single artefact, there is a wide range of possible presentation scenarios. Users might pick one among many artefacts from a shelf, or from a digital replica of the museum room, or even from a historic scene digitally rebuilt in order to contextualize the cultural artefacts by showing them in their historic sur-

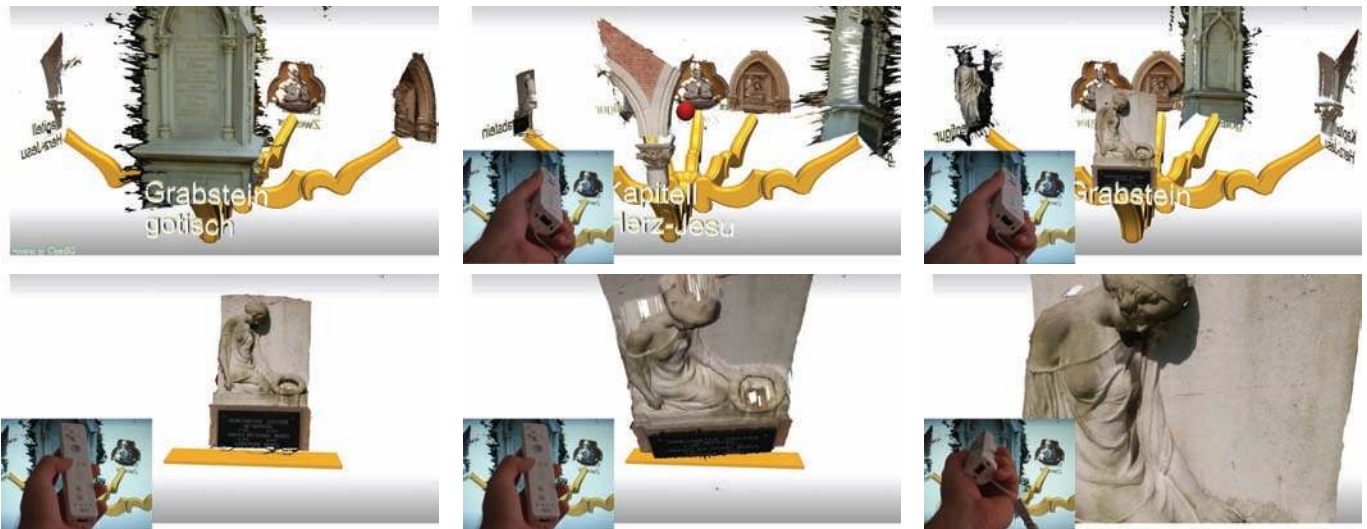


Figure 15: Example of a 3D presentation. Upper row: The user can rotate the chandelier horizontally. Lower row: One object is chosen for inspection, and the user can orbit around it. – The objects shown are not high-quality artefacts but only examples.

roundings. This flexibility shall become possible through *customizable presentation templates*, very basic 3D scenes with objects that have a reasonable pre-defined behaviour and whose appearance (geometry+texture) can be configured via drag-and-drop. Curators can download these presentation templates from suitable websites.

3D modeling of ad-hoc geometry: Sometimes ad-hoc objects are needed for a presentation. Static objects could be created photographically from digital photographs using the Epoch Webservice [VG06]. However, this is not applicable in all cases. To let users generate simple objects, e.g., extruded 2D contours, a very simple 3D modeling tool should be part of the authoring software. This tool is comparable to the vector-based diagram editor included in, e.g., Microsoft Powerpoint. And just like Powerpoint it should allow to animate these diagrams by animating the object parameters. This way a 3D stone wall could vary its position and (x, y, z) -extents very much like a line segment in a 2D drawing can.

Non-monolithic: From a software point of view the presentation viewer shall behave like a component, rather than like a stand-alone application. The reason is re-usability: Along with the 3D presentation additional textual information might have to be displayed, a HTML page or a pdf document. It shall even be possible to integrate the 3D presentation with another application that has its own GUI, such as a numeric simulation or a database front-end. The consequence is that the 3D presentation viewer shall require not much more than a 3D window to render into; another consequence is that it does not provide a sophisticated 2D GUI with a menu hierarchy (like MS Powerpoint has). It should be possible, though, to later add a 2D GUI with a menu.

Developer Levels: We envisage a hierarchy of users of our systems. Each level requires more knowledge and, thus, will reach out to a smaller community:

- Level 0:** End-user who consume the 3D-presentations
- Level 1:** Authoring of presentations: 3D-GUI, drag&drop
- Level 2:** Authoring of presentation templates: Scripting

Level 3: Extension developers: C++ programming

Users on levels 1-3 are creative people, the DCC providers, which stands for *digital content creation*.

6. The Epoch Viewer

Our solution is based on the combination of the OpenSG scene graph system with the GML scripting language [Hav05].

We have developed a series of GML scripts for 3D modeling, for presentation templates, and for particular presentations. The first phase of our work concentrated on providing OpenSG with the functionality needed, and on making it accessible via GML in a suitable fashion. Next we have begun to create a number of example presentations on this basis. The next step, which will start soon, is to revise and refactor the GML code for those presentations. The goal is to distill a set of basic GML components out of these presentations, in order to produce a GML framework that will be useful for all sorts of presentations. The final step will be to create a conventional GUI and menu system, which makes use of the 3D presentation as a component.

6.1. Input device: The tangible proxy object

By far the most intuitive 3D input device is a 3D object. The idea of the *proxy object* is that the virtual object displayed on the screen moves exactly like the real object that the user holds in his hands. Ideally, the user can move and rotate the object, and the digital artefact is in exact sync. Note, however, that we want to map the 6-DOF pose directly, not in a mirror fashion, so that when the user stretches out the hand with the proxy object the virtual object also goes farther away. It does not come closer as would be the case with a mirror. – Technically, the problem is to determine the 6-DOF pose of the proxy object. We have experimented with the three technologies shown in Fig. 14.

First technology: ARToolkit. We have tried camera-based tracking using the ARToolkit from www.artoolkit.org. With one marker



Figure 16: Authoring a presentation with BTF-models. They are loaded and rendered using an OpenSG extension module that makes use of advanced vertex and pixel shaders. Models are courtesy Gero Müller and Reinhard Klein, Univ. Bonn, Germany

per side of a cube and a single camera we had serious robustness problems: Whenever a finger only *touched* the black boundary of the marker the recognition algorithm of ARToolkit broke. Consequently we have made the cube a bit larger and used an array of 2×2 markers. This seems to be a good compromise between robustness and processing load, as the latter affects the recognition speed and, thus, the frame rate. It is quite unlikely that the user occludes all four markers of one side at the same time, and usually more than one side is visible.

Second technology: Accelerometer. ARToolkit markers have a very technical appearance which we wanted to avoid. The position tracking also created slight problems since users tended to move the 3D object out of the frustum: To inspect the object they took it very closely until the view showed only a detail of the surface; but then, in order to rotate it, they took it from one hand to the other, thereby actually moving it quite a bit to the right or the left. This made the object suddenly disappear, and the users got lost and felt uncomfortable struggling to bring the object back.

Consequently we tried to get along by using only the orientation (acceleration) information. With an object held still, gravity causes the strongest acceleration, so the downwards direction can be robustly detected. Roll information around the gravity vector, though, can not be detected, so it can not be decided whether the user points north, east, or west. So we used the accelerometer information only for relative motion (spinning speed). This worked well and robustly.

The accelerometer is a standard electronic device and quite cheap (15 Euros). It can easily be connected to a serial or USB port. Due to its small size it can also be put inside another object, e.g., one that resembles a cultural artefact. This looks much better in a museum than ARToolkit markers.

Third technology: Nintendo Wiimote. The controller of the Nintendo Wii, the *Wiimote*, communicates with standard Bluetooth. Free software tools exist to decode its protocol, e.g., Kenner's *GlovePIE* [Ken]. The Wii can deliver also position information, as it contains an optical sensor that, when pointing towards a certain configuration of LEDs, determines the pose relative to the LEDs. The Wiimote is a mature device and quite robust to use, which made it our preferred test device, despite its non-museal appearance.

6.2. The 3D Presentation

A first example of a 3D presentation is shown in Fig. 15. The user sees a nobject selection menu that is shaped like a chandelier. With a slight rotation of the Wiimote to the left or the right the chandelier begins as well to rotate smoothly, showing the next object in

the respective direction. By tilting the Wiimote upwards the close-up view is activated: The chandelier gradually moves away and the chosen object comes close until it fills the view.

One of our goals was that users can always keep track of what is going on. There are no abrupt transitions and we have taken care that all motions are smoothly animated. Before the chandelier moves away, it retracts; when the close-up inspection is finished, the chandelier appears again and unfolds, see Fig. 18. The close-up object can be inspected in detail: With the two DOFs of the Wiimote (rotate L/R and U/D, for left/right, up/down) it is only possible to orbit around the object center in a fixed distance: In terms of Euler angles, L/R determines the azimuth and U/D the elevation of the object.

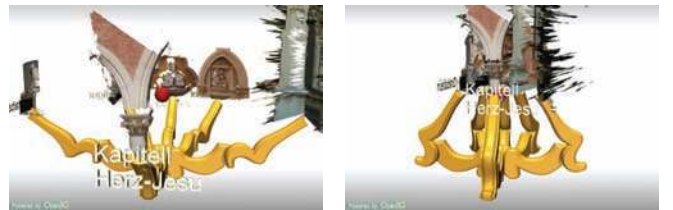


Figure 18: The “chandelier” smoothly retracts before moving away, as a clear signal that close-up inspection begins

We have experimented also with a combined interaction mode: The elevation must be clamped, e.g., to $[-70, 70]$ degrees to avoid the gimbal lock. When the elevation is maximal or minimal, a further increase or decrease makes the object come closer or get farther away, respectively. – Users usually reported a good feeling of control in this combined mode. The problem was only that they found it uncomfortable: It is apparently more convenient, or natural, to first navigate (orbit) to a particular spot on the surface, and then to adjust the distance to this spot. Our mode required to first adjust the distance, and then orbit over the surface.

Note that the objects in Fig. 15 are low quality reconstructions, generated photogrammetrically from a single range map and decimated to 10K vertices using Michael Garland's quadric-based simplification tool *qslim*. A much higher rendering quality can be obtained using the BTF-rendering module for OpenSG from Gero Müller and Reinhard Klein (Univ. Bonn). A BTF provides much more surface detail as it approximates the BRDF with much more than only a single texture value per surface point. Especially small and shiny BTF surface parts are brought out by the headlight much more clearly. Fig. 16 can only deliver part of the experience to hold a shimmering object virtually in his own hands.

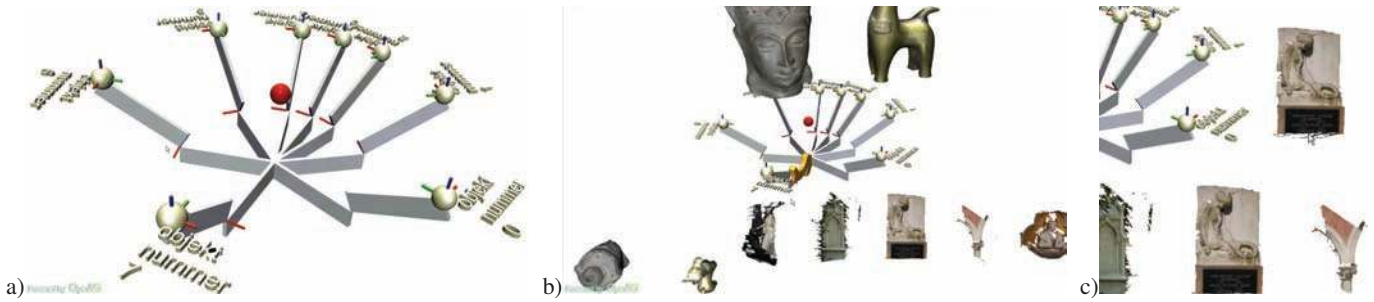


Figure 17: Authoring via drag-and-drop. a) A blank layout consists of many drop targets. b) The “model bar” is filled via drag-and-drop from the file system with 3D models, 2D images, and 1D text strings. c) Objects are dragged interactively from the model bar to drop targets in the 3D scene, where they automatically align to the local coordinate frame.

6.3. Authoring of a 3D Presentation



Figure 19: Presentation layout: Parametric shelf. All the boxes can be dragged to adjust the spacing, and pushing the red spheres inserts a new row or column of planks.



Figure 20: Presentation layout: Chandelier. Even when densely populated, the selected object sticks out clearly due to the uneven circular spacing.

We have experimented with several sorts of layouts. The first idea was a rack or shelf (Fig. 19) where the spacing can be interactively adjusted to match the sizes of the artefacts. Fig. 20 shows our chandelier-like design. Its special feature is that it rotates non-linearly in order to clearly highlight the object that can be chosen for the detailed inspection.

6.4. Authoring via Drag & Drop

The main idea is that our layouts are almost completely composed of so-called *drop targets*. Fig. 17a shows such a “blank” layout. All the boxes and spheres are successively replaced. Three different types of objects are supported: **3D models** (Collada .dae, Wavefront .obj, Stanford .ply, OpenSG .osb, etc), **images** (.png or .jpg, immediately applied to texture a quadrangle), and **character strings**, which are rendered as true 3D text. The replacement proceeds in two steps:

- **Filling the model bar:** The user drags an object from the file system (the Windows Explorer) to the 3D window where it appears in a row along the lower border of the 3D window, the *model bar* (Fig. 17b)
- **Replacing the drop targets:** Individual models can be dragged interactively from the model bar to drop targets in the 3D scene (Fig. 17c). Whenever dragging the object over a suitable drop target the object temporarily snaps and aligns with this target. The user can decide whether to leave it there (mouse release), or continue to drag it elsewhere.

Note that objects can not be dragged immediately from the file system to a drop target in the scene; the model bar always acts as an intermediate storage. The reason is that with larger models there is a noticeable delay until the object is loaded and appears in 3D under the mouse pointer. Users would instinctively think that the dragging operation has failed, and stop dragging – only to see that after a short while the object appears *somewhere* in the scene. This was perceived so frustrating that we decided to introduce the model bar.

Another thing that has proven successful was that when loading an object we immediately show a temporary geometry, a sphere, that is replaced by the true object as soon as its loading is finished. We do not, however, use the sphere for immediate drag-and-drop because of size issues: The temporary sphere can not reflect the true size of the object that is being loaded, simply because the bounding box of this object is only available after it has been loaded.

We plan to solve this issue using the so-called *Collada light* 3D format: The (lightweight) Collada XML file contains only semantic and metadata information, in particular the bounding box, and it references another (heavy) binary file that contains the 3D data, for instance a huge U3D file with compressed triangle data.

6.5. Modeling Included: Creating Ad-Hoc Geometry

The design of the arms of the chandelier was based on a small sketch on a piece of paper that, in the museum scenario, would have come from a curator or from a graphics designer (see Fig. 21). The sketch was photographed, the photograph was perspective corrected by marking four points and then it was loaded into the GML based modeller. The modelling proceeds in a very simple rubber-band fashion using a control polygon (Fig. 23): Clicking on the polygon boundary inserts a new ball. Balls can be freely dragged around on the construction plane. Just clicking on a ball toggles its red/green status: green balls control a B-Spline, red balls are corners. Clicking

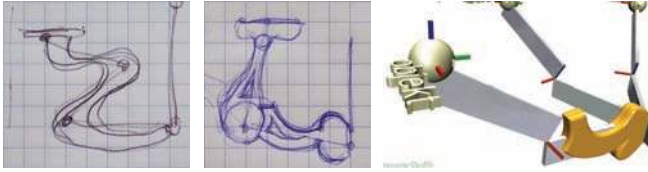


Figure 21: Two designs for the arm of the chandelier. The ad-hoc geometry created from the sketches is being dragged onto the drop target. As soon as the mouse button is released it is copied on all instances of the arm.

on the blue ball extrudes the polygon. The extrusion profile can also be adjusted, but this is not shown here. – The ad-hoc geometry thus created is then dragged into the authoring toolkit.

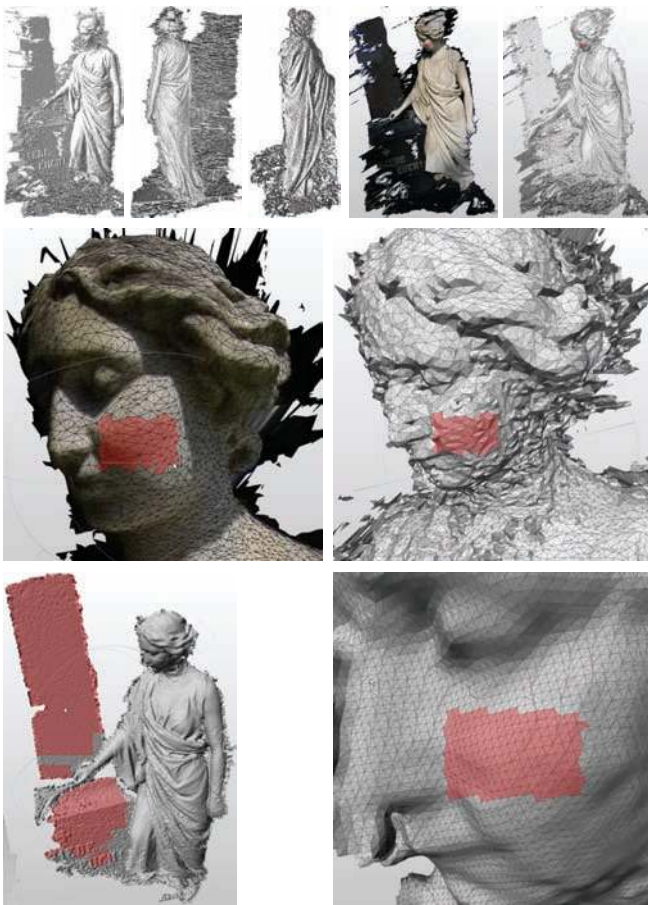


Figure 22: The shape markup problem. The *cheeks* of the statue are a concept that exists independently from the particular shape representation that is used. The problem is how to store the shape markup in a sustainable way such that it is robust against remeshing and mesh processing of the various input data (first row) that are used to obtain the final model (bottom right).

7. Conclusion and Future Work

The Epoch viewer presented in this paper leaves room for further improvement; in fact, it is still work in progress, since by far not all

features from the feature wish list could be realized. But although our results are far from perfect we claim that the foundation is sound with respect to the feature wish list from section 5, with which we would also highlight some areas for further research:

- **Drag-and-Drop Authoring:** The drop targets work extremely well in the authoring mode. Future layouts will have also drop targets for images (backdrop) and text. The model bar should be part of a 2D GUI, though.
- **Versatile Hardware Setups:** Any input device that uses the channel concept can connect via socket. However, this is not a solution for input devices that require very much bandwidth such as, e.g., cameras.
- **Easy 3D navigation:** The tradeoff between comfort and control must be improved further by employing more intelligent camera behaviours. 3D navigation in the authoring mode must be improved as well.
- **CH Integration:** As soon as Collada files offer more standard (meta-)information, this information should be available for 3D presentations. Examples: The textual caption of the 3D model, and information links embedded in the surface.
- **CH Sustainability:** Clearer separation between framework and individual layout, so that the layout style of a presentation can be exchanged like in MS Powerpoint.
- **Low-Level extensible:** Accomplished in principle, but unclear for visualisations that are genuinely 2D.
- **Cuneiform viewer module:** This is genuinely 2D and, thus, still open.
- **Template extensible:** Accomplished in principle but now needs many more examples
- **3D modeling of ad-hoc geometry:** It is unclear how sophisticated the 3D modeling should be: Should it be like Google Sketchup or significantly less powerful? – Very interesting though would be the use of animated parametrized models as 3D-illustrations.
- **Non-monolithic:** This is definitely accomplished. The presentation viewer can in fact be embedded as a component into any application providing an OpenGL window. All program functions can be accessed through GML scripts that can even be synthesized at runtime.
- **Developer Levels:** A proper documentation of the scripting facility is the only thing that is required for others to develop interesting presentation templates.

The remaining problems can be seen as symptoms of underlying much more fundamental issues (that are alluded to in Fig. 22). The articles on the *seven open research questions* [HF07, BFH*07] for example identify some areas where radically new solutions and approaches need to be developed before we are able to manage really large numbers of valuable Cultural Heritage assets in digital libraries reliably and sustainably.

Acknowledgements

We would like to thank prof. Karel Van Lerberghe and Hendrik Hameeuw for the given insight into the study of cuneiform tablets and the extended dissemination of the acquisition system.

References

- [Ben07] BENDELS G. H.: *Methods for 3D Geometry Processing in the Cultural Heritage Domain*. PhD thesis, University of Bonn, Germany, 2007.

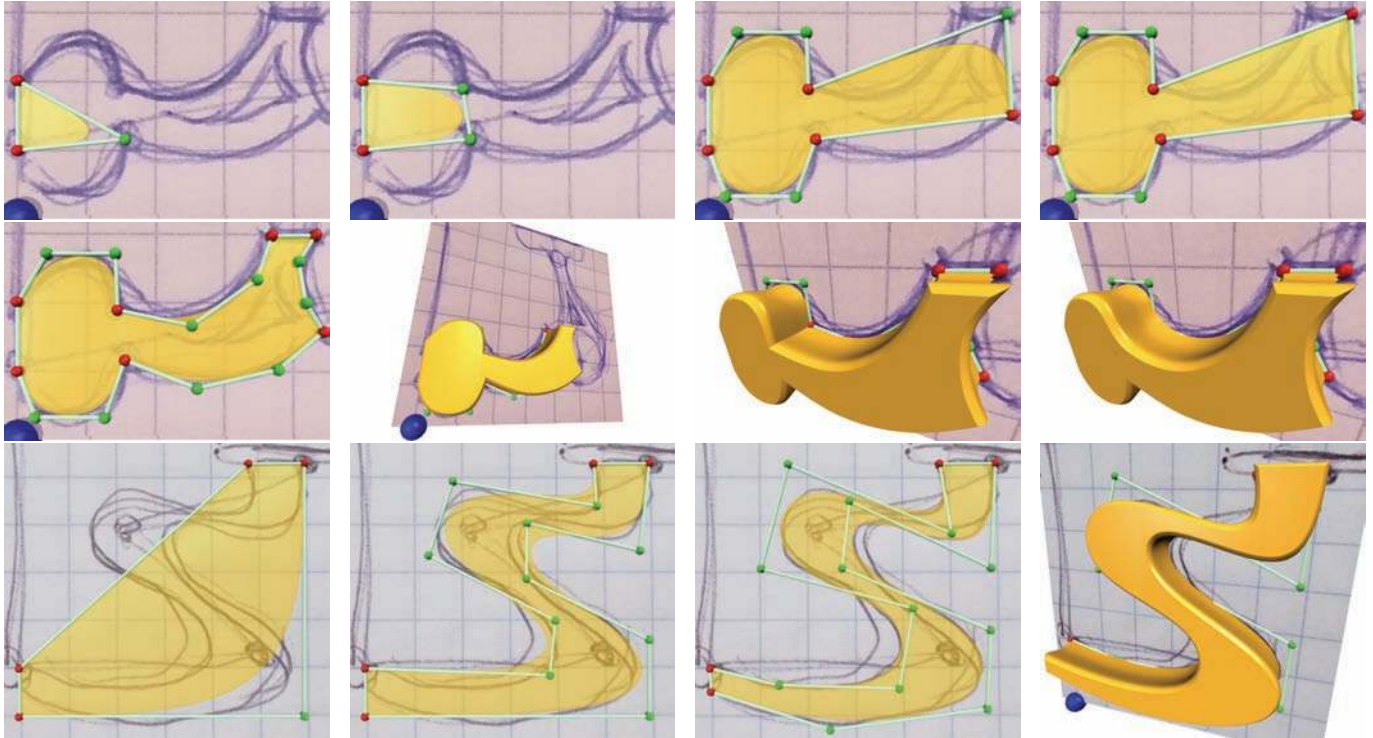


Figure 23: Creating ad-hoc geometry from a sketch. Row 1: A rubber band triangle is successively expanded. Row 2: The profile is extruded, polygon vertices can be sharp (B-spline) or smooth (corner). Row 3: Most users understand quickly how to place vertices in a way such that the resulting raw shape can be adjusted efficiently.

[BFH*07] BUSTOS B., FELLNER D., HAVEMANN S., KEIM D., SAUPE D., SCHRECK T.: Foundations of 3d digital libraries: Current approaches and urgent research challenges. In *Proceedings of the First International Workshop on Digital Libraries Foundations (DLF1), Vancouver, British Columbia, Canada, June 23, 2007* (2007), DELOS Network of Excellence on Digital Libraries.

[BSly] BBC, SILICON GRAPHICS (SGI): Mummy: Inside story, 2004 July. Exhibition at the British Museum London, <http://www.thebritishmuseum.ac.uk>.

[CGS06] CHANG T., GOESELE M., SEIDEL H.-P.: Mesostucture from specularly. In *CVPR* (2006), pp. 1825–1832.

[CMFP02] COSTAGLIOLA G., MARTINO S. D., FERRUCCI F., PITTARELLO F.: An approach for authoring 3d cultural heritage exhibitions on the web. In *SEKE '02: Proc. 14th int'l conference on Software and knowledge engineering* (New York, NY, USA, 2002), ACM Press, pp. 601–608.

[CPCS06] CALLIERI M., PONCHIO F., CIGNONI P., SCOPIGNO R.: Easy access to huge 3d models of works of art. In *Fourth Eurographics Italian Chapter 2006* (Graz, Austria, 2006), Fellner D., (Ed.), Graz University of Technology, Eurographics Association, pp. 29–36. Catania, 22-24 February 2006.

[CPCS08] CALLIERI M., PONCHIO F., CIGNONI P., SCOPIGNO R.: Virtual inspector: a flexible visualizer for dense 3d scanned models. *IEEE Computer Graphics and Applications* 28, 1 (Jan.-Febr. 2008), ??–??

[DMK03] DEGENER P., MESETH J., KLEIN R.: An adaptable surface parametrization method. In *The 12th International Meshing Roundtable 2003* (September 2003).

[DNvGK97] DANA K., NAYAR S., VAN GINNEKEN B., KOENDERINK J.: Reflectance and texture of real-world surfaces. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings* (1997), 151–157.

[epi07] EPISODE E. F.: European digital library to contain all knowledge. (Belgian television), 2007. www.euronews.net/index.php?page=futuris&article=444863.

[Hav05] HAVEMANN S.: *Generative Mesh Modeling*. PhD thesis, Institute of Computer Graphics, Faculty of Computer Science, Technical University Braunschweig, Germany, November 2005.

[HBD07] HAHN D. V., BALDWIN K. C., DUNCAN D. D.: Non-laser-based scanner for three-dimensional digitization of historical artifacts. *Applied Optics* 46, 15 (May 2007), 2838–2850.

[HF04] HAVEMANN S., FELLNER D. W.: Generative parametric design of gothic window tracery. In *Proc. VAST 2004: The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage* (Brussels, Belgium, 2004), Cain K., Chrysanthou Y., Niccolucci F., Silberman N., (Eds.), Eurographics, pp. 193–201.

[HF07] HAVEMANN S., FELLNER D. W.: Seven research challenges of generalized 3d documents. *IEEE Computer Graphics and Applications* 27, 3 (2007), 70–76. (special issue on 3D documents).

[HS03] HERNÁNDEZ C., SCHMITT F.: Silhouette and stereo fusion for 3d object modeling. In *3DIM 2003, 4th International*

- Conference on 3D Digital Imaging and Modeling (2003), pp. 46–53.
- [HSKF06] HAVEMANN S., SETTGAST V., KROTTMAIER H., FELLNER D.: On the integration of 3d models into digital cultural heritage libraries. In *VAST 2006 Project Papers* (Nicosia, Cyprus, Nov 2006), Ioannides, Arnold, Niccolucci, Mania, (Eds.), Epoch public., pp. 161–169.
- [Ken] KENNER C.: Glove programmable input emulator (glovepie). carl.kenner.googlepages.com/glovepie.
- [Lau94] LAURENTINI A.: The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* 16, 2 (1994), 150–162.
- [MBK05] MÜLLER G., BENDELS G. H., KLEIN R.: Rapid synchronous acquisition of geometry and btf for cultural heritage artefacts. In *The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)* (November 2005), Eurographics Association, Eurographics Association, pp. 13–20.
- [MBR*00] MATUSIK W., BUEHLER C., RASKAR R., GORTLER S. J., MCMILLAN L.: Image-based visual hulls. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 369–374.
- [MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 519–528.
- [MMK04] MÜLLER G., MESETH J., KLEIN R.: Fast environmental lighting for local-pca encoded btfs. In *Computer Graphics International 2004 (CGI 2004)* (June 2004), IEEE Computer Society, pp. 198–205.
- [MSK06] MÜLLER G., SARLETTE R., KLEIN R.: Data-driven local coordinate systems for image-based rendering. *Computer Graphics Forum* 25, 3 (September 2006).
- [MVSL05] MUDGE M., VOUTAZ J.-P., SCHROER C., LUM M.: Reflection Transformation Imaging and Virtual Representations of Coins from the Hospice of the Grand St. Bernard . In *The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST* (Pisa, Italy, 2005), Mudge M., Ryan N., Scopigno R., (Eds.), Eurographics Association, pp. 29–39.
- [MVW*06] MÜLLER P., VEREENOGHE T., WONKA P., PAAP I., GOOL L. V.: Procedural 3D Reconstruction of Puuc Buildings in Xkipché. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage* (Nicosia, Cyprus, 2006), Ioannides M., Arnold D., Niccolucci F., Mania K., (Eds.), Eurographics Association, pp. 139–146.
- [MWGA06] MALZBENDER T., WILBURN B., GELB D., AMBRISCO B.: Surface Enhancement Using Real-time Photometric Stereo and Reflectance Transformation. In *Symposium on Rendering* (Nicosia, Cyprus, 2006), Eurographics Association, pp. 245–250.
- [VG06] VERGAUWEN M., GOOL L. V.: Web-based 3d reconstruction service. *Machine Vision Applications*, 17 (2006), 411–426.
- [VWV06] VERBIEST F., WILLEMS G., VAN GOOL L.: Image-based rendering for photo-realistic visualization. *Virtual and Physical Prototyping (Taylor & Francis) I* (2006), 19–30.