

# 3D-Powerpoint – Towards a design tool for digital exhibitions of cultural artifacts

S. Havemann<sup>1</sup>, V. Settgast<sup>1</sup>, M. Lancelle<sup>1</sup>, D.W. Fellner<sup>1,2</sup>

<sup>1</sup> Institute of Computer Graphics and Knowledge Visualization (CGV), TU Graz, Austria

<sup>2</sup> GRIS, TU Darmstadt & Fraunhofer IGD, Germany



---

## Abstract

We describe first steps towards a suite of tools for CH professionals to set up and run digital exhibitions of cultural 3D artifacts in museums. Both the authoring and the presentation views shall finally be as easy to use as, e.g., Microsoft Powerpoint. But instead of separated slides our tool uses pre-defined 3D scenes, called "layouts", containing geometric objects acting as placeholders, called "drop targets". They can be replaced quite easily, in a drag-and-drop fashion, by digitized 3D models, and also by text and images, to customize and adapt a digital exhibition to the style of the real museum. Furthermore, the tool set contains easy-to-use tools for the rapid 3D modeling of simple geometry and for the alignment of given models to a common coordinate system.

The technical innovation is that the tool set is not a monolithic application. Instead it is completely based on scripted designs, using the OpenSG scene graph engine and the GML scripting language. This makes it extremely flexible: Anybody capable of drag-and-drop can design 3D exhibitions. Anybody capable of GML scripting can create new designs. And finally, we claim that the presentation setup of our designs is "grandparent-compliant", meaning that it permits to the public audience the detailed inspection of beautiful cultural 3D objects without getting lost or feeling uncomfortable.

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

---

## 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 arbitrary visitors take them in their hands. It is particularly difficult to let visitors from the general public study the amazing, intricate detail and the traces a long history has left on cultural artifacts.

3D technology can help tremendously to enhance the appreciation of museum objects. Our guiding vision is the idea

of a *complementary exhibition*: Real artifacts in a museum exhibition are complemented by digital artifacts whose sole purpose is to deepen the understanding and appreciation of the real ones. A particular form of a digital artifact 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, which have greatly increased over the last years. School children are en-

couraged 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 visitor's attention from the artifact.

### 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 artifact in the field to the presentation of the artifact 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 artifacts were found, as well as laser scans of the remains of houses, pillars, statues, etc. He puts all the digital artifacts 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 artifacts, as usual behind glass. But in every exhibition hall there is also a 3D multimedia terminal where the visitor can inspect the small beautiful artifacts interactively from all sides, which is not possible with the real. The visitor can also have a quick look at the archeological site where the artifact was found, which also brings up a few statues and some interesting architectural ornaments. The latter are shown in the form of high-resolution 3D scans.

### 1.3. Contribution of this Paper

The contribution of this paper is to present one part of the target scenario as work in progress: The software infrastructure that allows to design complementary exhibitions, and to present them to the public. Note that it has some features in common with Microsoft 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.

Since our's is a 3D tool we chose the name *3D-Powerpoint* as the working title for the project. Before starting the project we have made some research on tools to use. None of them fulfilled our requirements. This lead us to a refined list of requirements, our feature wish list in section 3.

## 2. Related Work

The presentation of 3D objects to a public audience is often considered a solved problem since a number of 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 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.

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](http://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 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.

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

### 3. Feature Wish List

**Drag-and-Drop Authoring:** Every presentation consists of *layout* and *content*. Both is 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.

**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

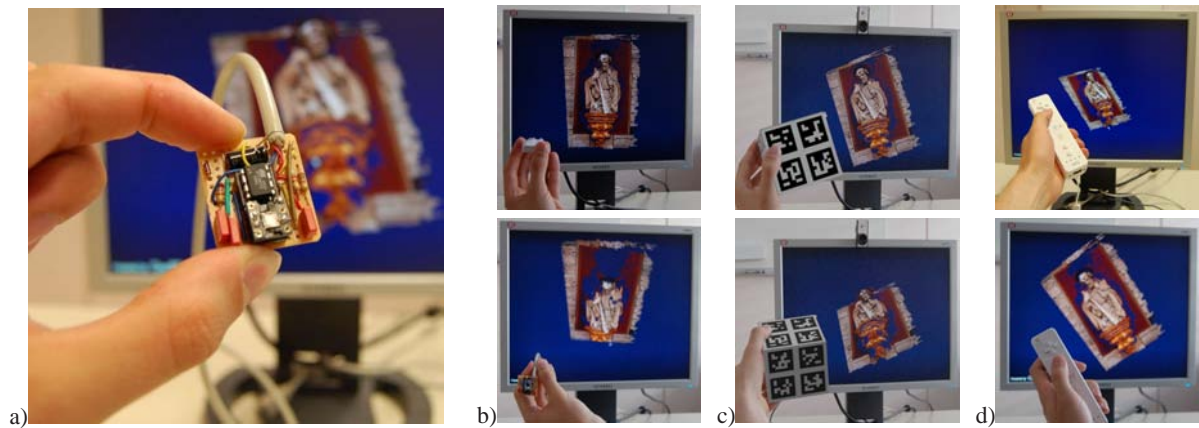
long run, even using CIDOC/CRM should be an option, as pointed out by Havemann et al. in [HSKF06]: Every cultural artifact is part of a network of semantic information. The ultimate form of a *Cultural Presentation Browser* would be a tool that allows average museum visitors to navigate through this 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 artifacts 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 artifacts 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.

**Template extensible:** Whereas the main use of the 3D kiosk is to let a visitor explore one particular artifact, there is a wide range of possible presentation scenarios. Users might pick one among many artifacts 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 artifacts by showing them in their historic surroundings. 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. It is envisaged that curators can download presentation templates from suitable web pages.

**3D modeling of ad-hoc geometry:** Sometimes ad-hoc objects are needed for a presentation. Static objects could be created photogrammetrically 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.



**Figure 1:** 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  $2 \times 2$  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.

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*.

#### 4. A First Version of 3D-Powerpoint

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

We have developed a series of GML scripts for 3D mod-

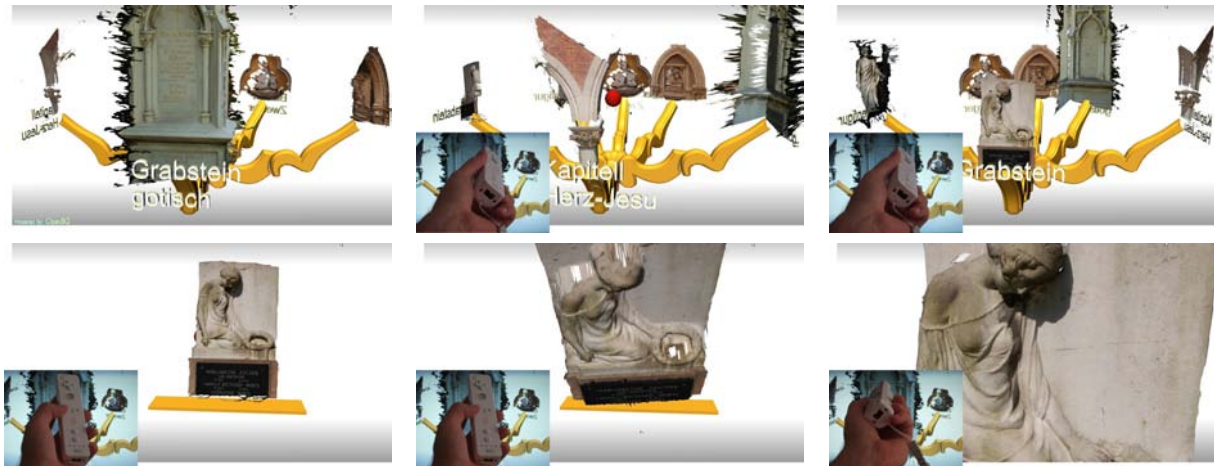
eling, for presentation templates, and for particular presentations. The first phase of our work concentrated on providing OpenGL with the functionality needed, and on making it accessible via GML in a suitable fashion. Next we have begun to create a number of example presentations on this basis. The 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.

#### 4.1. 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 artifact 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. 1.

**First technology: ARToolkit.** We have tried camera-based tracking using the ARToolkit from [www.artoolkit.org](http://www.artoolkit.org). With one marker 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





**Figure 2:** 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.

bit larger and used an array of  $2 \times 2$  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 artifact. 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-museum appearance.

#### 4.2. The 3D Presentation

A first example of a 3D presentation is shown in Fig. 2. The user sees a object 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. 3.

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.

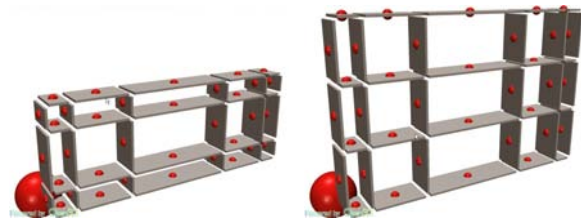
We have experimented also with a combined interaction mode: The elevation must be clamped, e.g., to  $[-70, 70]$  degrees to avoid the gimbal lock. When the elevation is maximal or minimal, a further increase or decrease makes the



**Figure 4:** 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



**Figure 3:** The “chandelier” smoothly retracts before moving away, as a clear signal that close-up inspection begins



**Figure 5:** 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.

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. 2 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. 4 can only deliver part of the experience to hold a shimmering object virtually in his own hands.

#### 4.3. Authoring of a 3D Presentation

We have experimented with several sorts of layouts. The first idea was a rack or shelf (Fig. 5) where the spacing can be interactively adjusted to match the sizes of the artifacts. Fig. 6 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.

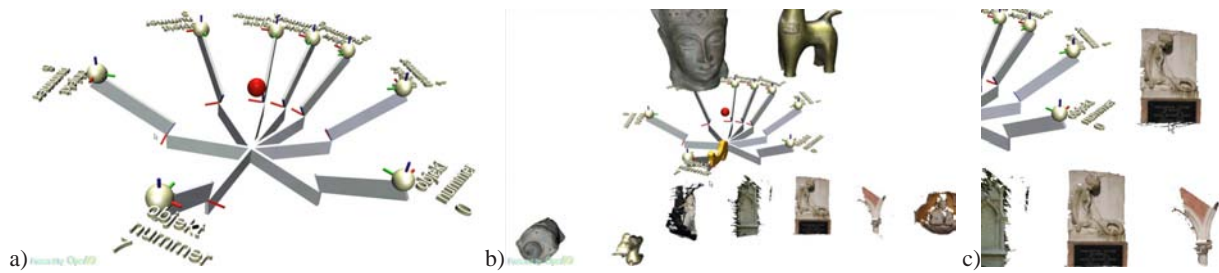


**Figure 6:** Presentation layout: Chandelier. Even when densely populated, the selected object sticks out clearly due to the uneven circular spacing.

#### 4.4. Authoring via Drag & Drop

The main idea is that our layouts are almost completely composed of so-called *drop targets*. Fig. 7a 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. 7b)
- **Replacing the drop targets:** Individual models can be dragged interactively from the model bar to drop targets in the 3D scene (Fig. 7c). Whenever dragging the object over a suitable drop target the object temporarily snaps



**Figure 7:** 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.

and aligns with this target. The user can decide whether to leave it there (mouse release), or continue to drag it elsewhere.

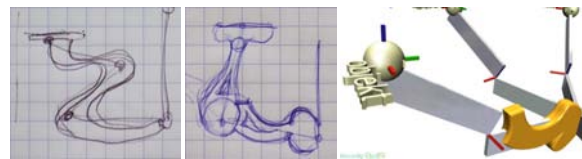
Note that objects can not be dragged immediately from the file system to a drop target in the scene; the model bar always acts as an intermediate storage. The reason is that with larger models there is a noticeable delay until the object is loaded and appears in 3D under the mouse pointer. Users would 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.

#### 4.5. 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. 9). 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. 8): 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



**Figure 9:** 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.

balls are corners. Clicking on the blue ball extrudes the polygon. The extrusion profile can also be adjusted, but this is not shown here.

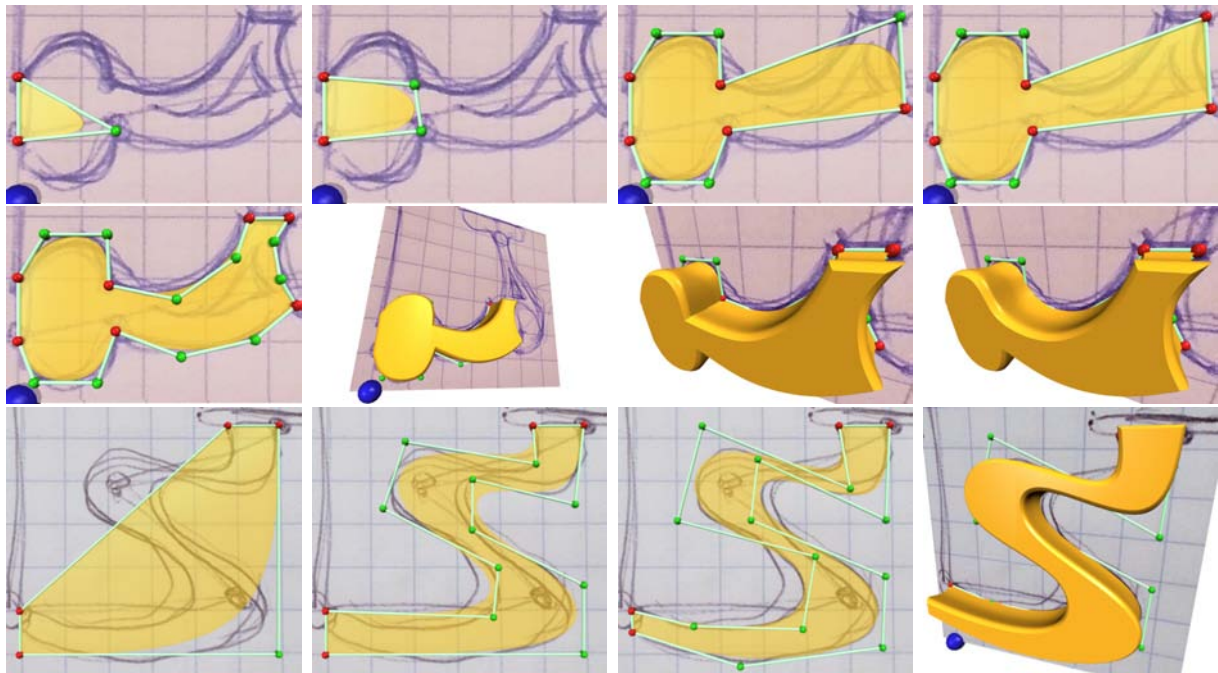
The ad-hoc geometry thus created is then dragged into the authoring toolkit.

#### 5. Conclusion and Future Work

The 3D-Powerpoint toolkit presented in this paper leaves much room for improvement; in fact, it is work in progress. Although our results are far from perfect we claim that the foundation is sound with respect to the feature wish list from section 3, 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.
- **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.





**Figure 8:** 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.

- **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:** This is accomplished.
- **Template extensible:** This is accomplished as well.
- **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.

## References

- [BSly] BBC, SILICON GRAPHICS (SGI): Mummy: Inside story, 2004 July. Exhibition at the British Museum London, <http://www.thebritishmuseum.ac.uk>.
- [CMFP02] COSTAGLIOLA G., MARTINO S. D., FER- RUCCI 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, 2006. Catania, 22-24 February 2006.
- [CPCS08] CALLIERI M., PONCHIO F., CIGNONI P., SCOPIGNO R.: Virtual inspector: a flexible visualizer for dense 3d scanned models, Jan.-Febr. 2008.
- [Hav05] HAVEMANN S.: *Generative Mesh Modeling*. PhD thesis, Institute of Computer Graphics, Faculty of Computer Science, Technical University Braunschweig, Germany, November 2005.
- [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](http://carl.kenner.googlepages.com/glovepie).
- [VG06] VERGAUWEN M., GOOL L. V.: Web-based 3d reconstruction service. *Machine Vision Applications*, 17 (2006), 411–426.