Methods and application of interactive 3D computer graphics in anthropology

Stefan Seipel and Mikael Lindkvist

Department of Information Technology
Uppsala University
Box 337, SE-751 05 Uppsala, Sweden

Technical report 2002-002
January 2002
ISSN 1404-3203

Methods and application of interactive 3D computer graphics in anthropology

Seipel S. and Lindkvist M.

Human-Computer Interaction

Department of Information Technology

Uppsala University

Abstract

This report presents 3D computer graphics techniques for interactive visual exploration of virtual mummies. It is part of an exposition on the Cultural Heritage of the Egyptian Mummies in the Museum Gustavianum, in Uppsala. We describe a general-purpose projection metaphor for correctly presenting virtual 3D images on the dissection table of a historical anatomic theatre. This method allows for dynamic off-axis perspective viewing situations as well as it provides keystone correction for excessive projection angles as necessitated by the specific installation environment. For the application to reach out beyond the scope of the exhibition, we developed an adaptive image-based rendering approach that scales with the performance of the rendering host. Based on dynamic mesh simplification of the 3D mummy model, it automatically performs re-projections of texture images in order to maintain correct visual results. For interaction purposes with a digitiser tablet we present a means of stroke-based input that provides ease of use to non-expert visitors of the exhibition.

The problems we address with this paper are not only of interest for this particular application domain but generally for all interactive graphical installations, which must be adapted to the existing architectural situation.

Keywords: Viewing metaphors, orthoscopic projection, keystone correction, texture mapping, 3D interaction, virtual reality, anthropology.

1. Introduction

Methods of computer graphics have during the last decade been successfully applied to reconstructions and visualisations of archaeological artefacts. In anthropological research, non-destructive methods as e.g. three-dimensional computed tomography (CT) have created new opportunities to perform studies on precise details of the mummification procedures without destructing invaluable ancient mummies. Most of the paleopathological research presented earlier in this context has focussed on the data acquisition process by using spiral CT [1,2,3]. Recent research has been carried out in post-processing algorithms and tissue classification to enhance visual quality of 3D reconstruction results and to enhance identification of morphological structures [4]. Almost all of these previous efforts have led to methods and tools that are used by experts and researchers in the scientific community. As such, the developed applications are designed to support numerical analysis and scientific visualization of the vast amount of data collected by the various imaging modalities.

In the context of a museum exhibition on ancient Egyptian mummies at the Museum Gustavianum in Uppsala, Sweden, it was the intention to make interactive graphical methods for exploration of virtual mummies accessible to the public. The application should allow for the visitors of the exhibition to study internal structures and artefacts beyond the surface of the linen wrappings. What renders the installation of the virtual mummy application specific is its historical surrounding: The Gustavianum building is among the most distinctive historic buildings in Uppsala. In the middle of the 17th century, the famous professor Olov Rudbeck had an impressive cupola built on top of the Gustavianum. The cupola was housing an anatomic theatre where dissections were to take place at that time. In 1715, this anatomic theatre was hosting the first historically documented exhibition of a mummy in Sweden; and by the end of the 19th century it was popular to perform dissections of Egyptian mummies in these kind of anatomic theatres throughout Europe. 250 years after the first dissection, and in the context of the Mummy Exhibition, virtual dissections were again intended in this particular historical setting.

Initially, we investigated the usability of several commercially available software solutions that would allow for exploration of 3D volumetric data sets of the mummy. Among those were e.g. Analyze^{TM} or Amira^{TM}, which both are commonly used tools in the scientific community for the purpose of data post-processing, segmentation and 3D surface reconstruction.

Very soon we realized that these tools were suitable to recreate 3D models for the virtual mummies from raw data (i.e. CT data) and also for rendering of predefined animation sequences. For the purpose of interactive dissections of a virtual mummy, however, we found those tools to be inappropriate for several reasons:

As professional tools, these programs provide a very broad set of powerful image processing operations. These operations and their parameterisation require very specific skills far above the level of most museum visitors. Also, the interaction modalities of those systems are designed for a typical professional desktop workstation situation. In other words, graphical output is presented in several windows on-screen and based on centric perspective projections. User interaction with virtual 3D objects is limited to rotation or scale of entire objects and does not allow for more advanced behaviour of the visualized objects. Also, data entry and interaction is facilitated using standard keyboard and mouse devices, which are no preferable choices for public computer installations. Due to these shortcomings we felt that a naturally appearing implementation of the virtual mummy into the authentic context of the anatomic theatre was only possible by building a customized visualisation application, which addressed the very requirements of the architectural environment and the museum visitors.

2. Material

The mummy under investigation is from a four years old girl child and it is dated to the period between 0 and 100 A.C. As is typical for the applied preparation technique, the viscera remained partially in place. This mummy belongs among the best-preserved mummies in

Sweden. It was examined with CT in the framework of the Uppsala Mummy Survey in spring 2001. CT scanning was performed at 1 mm slice intervals, which resulted in a number of 976 axial slices of the mummy. The image matrix had a resolution of 512 by 512 pixels. In total, the primary raw data comprised approximately 488 megabyte of binary data. These 3D image raw data were subsequently post-processed using commercially available software packages. Eventually, a re-sampled and filtered volume of the size 128x128x488 voxels could be obtained for the purpose of surface reconstruction. Three-dimensional polygonal surface models of relevant structures were reconstructed semi-interactively in local regions of interest, since fully automated segmentation procedures did not yield satisfactory reconstruction results for all regions of the mummy. The data-acquisition procedures were carried out by the radiology department of Uppsala University Hospital (UAS) while the Center for Image Analysis (CBA) of Uppsala University provided the basic set of polygon meshes for interactive visualization. A list of objects with their absolute and relative number of triangles respectively can be found in Tab. 1.

Object	**Triangles**	**Rate**
Skull	5000	4,02%
Arms	9996	8,04%
Leg_upper	4020	3,23%
Knee	1983	1,59%
Leg_lower	5360	4,31%
Chest	19964	16,05%
Spine	20000	16,08%
Liver	3000	2,41%
Heart	500	0,40%
Skin	14988	12,05%
Linnen	39544	31,80%
Total	**124355**	100,00%

Tab. 1. Geometric models of the virtual mummy and their absolute and relative polygon count.

For improvement of the visual quality of the virtual mummy photographic material of the real mummy was acquired. A number of digital photographs were taken from predefined directions by an archaeologist. These digital pictures show the mummy in one frontal and two orthogonal lateral views (Fig. 2). The approximate photographic parameters such as focal distance, camera direction, and lens zoom were known.

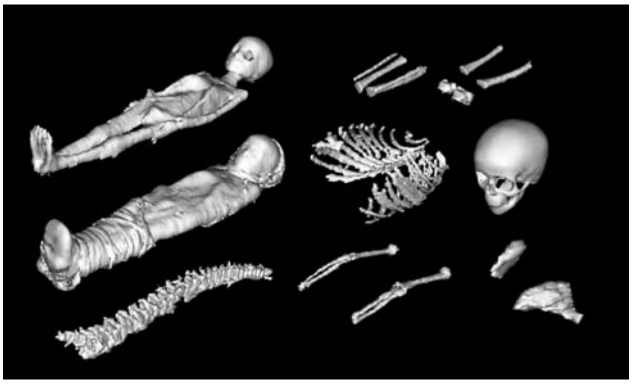

Fig. 1. Rendered pictures of the geometric models of the virtual mummy.

Fig. 2. Various views upon the real mummy were composed to one square format digital texture image.

For the presentation of the virtual mummy there were a number of specific pre-conditions: In the very centre of the anatomic theatre the historical dissection table is situated. It is elevated approximately 80 cm above the floor, and has a length of approximately 160 cm. The board of the table can be turned about 360 degrees. It is on top of the table were a front projection of virtual imagery was intended. Due to the historic value of this inventory, we could not permanently affix special screen material upon the table. Therefore we used fine woven white cotton linen to cover the dissection table (see also Fig. 3).

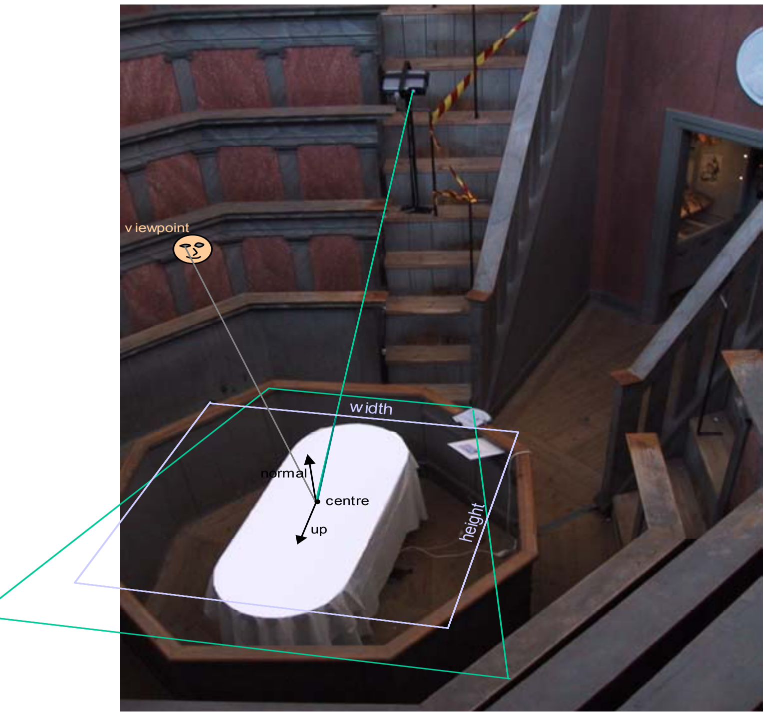

Fig. 3. The historic environment of the anatomic theatre where the virtual dissection is installed.

The historic environment put several constraints to the arrangement of the image projection source. In order to avoid distortions of projected images, it is generally desirable to locate the projector perpendicular to and above the display area (here the board of dissection table) or, if that is not possible, as close as possible to the central line of the projection. In the anatomic theatre, there is about 9 meters clearance above the dissection table up to the roof of the cupola, where a projector most suitably could be affixed. Though, several issues made this option impractical. The projection distance to the table would be very long in comparison to the relative small projection picture desired. This would call for a projector model with very long throw length, which are available, but at very high costs and to the expense of heavy weight. The latter implies also solid fixation points in the antique interior that are not feasible. Another aspect is that the entire structure of the cupola is made from wood. Wind under normal weather conditions introduces significant motion to the entire building structure of the cupola and hence to the projection image. Given all practical circumstances, the preferred solution for front projection was a low cost LCD projector with a native resolution of 1024 by 768 pixels mounted to a metal stud on the stairs upwards the gallery of the anatomic theatre. All involved people felt, that the solution as shown in Fig. 3 suited most inconspicuously into the historical environment of the anatomic theatre.

3. Methods

Starting from the initial situation described in the previous section, it was a challenging task to accomplish a realistic three-dimensional effect of the graphics depicted on the dissection table. Stereoscopic imagery is a strong visual 3D cue commonly used in virtual reality and interactive graphical systems [5]. It is relatively easy to implement in software, but due to the fragility of the required technical hardware stereoscopy is critical in the context of public installations. The chosen projection technique described above does generally not permit stereoscopic presentations. Alternative strong 3D cues that create strong illusions of three-dimensional imagery are motion parallax [6] and correct off-axis perspective projections [7]. The combination of both is often referred to as dynamic perspective displays and is often found in multiple screen VR environments [6] and horizontal stereoscopic displays [8, 9] that represent the same display conditions as in our application. In order to successfully implement viewpoint adapted perspective projections in the specific context, we needed to develop a projection metaphor that accommodates all parameters that render the actual display environment specific such as viewing position, off-axis projection angle, and size and orientation of the projection image.

3.1 A projection metaphor for keystone corrected off-axis 3D viewing

General methods for orthoscopic projection of 3D objects have been published before by different authors [10, 11]. While allowing for off-axis perspective projections, they make specific assumptions either in regard to the centre of projection or in regard to the orientation of the projection plane. Implementations of arbitrarily definable window-on-world viewing metaphors are meanwhile publicly available in a number of high level VR development libraries (e.g. CAVELib™, VR Juggler), but they do not support arbitrary manipulations of the projection matrix by the application developer.

As is indicated in Fig. 3, the position of the image projector creates a trapezoid distorted image in the projection plane resembled by the dissection table. In the actual situation, this distortion is excessive such that it cannot be compensated for by the built-in keystone correction functions of the projector. Therefore, the trapezoid projection image must be corrected by a 2D warping operation posterior to the 3D off-axis projection of virtual objects. This additional image transformation is part of the projection pipeline of the rendering software and must be implemented in the set-up procedure for the projection.

As is also evident from Fig. 3, the rotational orientation of the 2D window, which the virtual scenario is viewed through, is dictated by the allowable arrangement of the projector and projection screen (here the dissection table) in the physical environment.

Our viewing metaphor that is describing the 3D projection is a new notion of orthographic projections described similarly by other authors. In our notion the position and orientation of the projection window, the user's position, and the coordinates of virtual objects are expressed in terms of one common coordinate system, which is tied to a reference frame in physical space. We subsequently derive the parameterisation of the viewing metaphor such as to be implemented using projection concepts found in OpenGL:

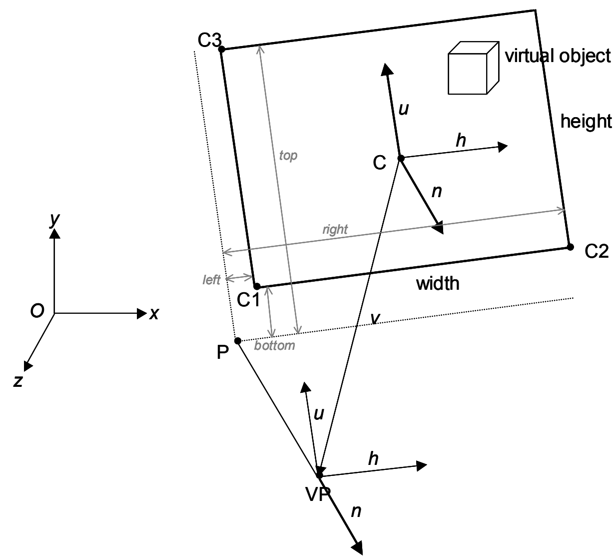

Fig. 4. Parameter definition of a viewing metaphor for arbitrary viewer-to-screen relationships.

Fig. 4 shows in a different picture the rectangular window through which objects in the scene are viewed. This window is defined by its centre C, a surface normal vector n, and a vector `u` that is pointing towards the direction of the vertical edge of the viewing window. In addition, the actual dimensions `width` and `height` of the projection window are given along with the position `VP` of the centre of projection (i.e. the eye position). All these parameters as well as the locations of virtual objects are denoted in real world metrics in relation to a common physical reference frame (O, x, y, z). In this sense, position and orientation of both physical objects (projection image on dissection table, observer) and virtual objects (the virtual mummy) can easily be co-located. Given the parameter set above to characterize an individual observer-screen relation, we can calculate `h`, a vector pointing along the horizontal direction of the window as follows:

$$h = u \times n \tag{1}$$

And, provided `n, h,` and `u` are normalized vectors, for the shortest distance `d` between VP and the plane that contains the window we obtain:

$$d = (VP - C) \bullet \vec{n} \tag{2}$$

An oblique projection of virtual objects in the scene upon the window towards the centre of projection VP can be accomplished through the *glFrustum()* concept of OpenGL that is well established. Two computational steps are required prior the actual projection using *glFrustum()*. First, the viewing frustum border parameters must be calculated based on the actual values `of width, height, C`, and `P`, which is the point where `VP` falls short off the plane containing the window. Where `P` can be calculated as:

$$P = VP - d \bullet \vec{n} \tag{3}$$

Given are also three corners of the viewing window as:

$$C1 = C - 0.5 \cdot \vec{h} - 0.5 \cdot \vec{u}$$
$$C2 = C + 0.5 \cdot \vec{h} - 0.5 \cdot \vec{u}$$
$$C3 = C - 0.5 \cdot \vec{h} + 0.5 \cdot \vec{u}$$

Then the border parameters of the viewing frustum `left, bottom, top` and `right` according to the drawing in Fig. 3 can be calculated as:

$$left = (C1 - P) \bullet \vec{h}$$
$$bottom = (C1 - P) \bullet \vec{u}$$
$$top = (C3 - P) \bullet \vec{u}$$
$$right = (C2 - P) \bullet \vec{h}$$

The *glFrustum* function assumes that the centre of projection is at `O={0,0,0}` and the projection plane is coplanar to `z=0` at some distance to `O`. In order to fulfil this precondition, the local viewing coordinate system `{VP, h, u, n}` must be transformed into the reference frame `{O, x, y, z}`. This operation affects conceptually the user defined viewing configuration and all other objects in the user scene. Therefore, this transformation operates on the modelview matrix stack. The matrix `T` that accomplishes the required transformation is implicitly given by the values of `VP, h, u,` and `n`:

$$T = \begin{bmatrix} h_x & u_x & n_x & VP_x \\ h_y & u_y & n_y & VP_y \\ h_z & u_z & n_z & VP_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$

The initialisation of the projection stack and model-view matrix stack can thereafter be performed straightforward according to OpenGL semantics as follows:

```
Compute d,T, left, bottom, top, right as above:
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
PostProjectionTransform();
glFrustum(left,right,bottom,top, d,
d+far_distance);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glMultMatrix(T);
```

So far, our projection metaphor provides an arbitrary observer-to-screen relationship and it is, apart from our specific parameterisation, conceptually not very different from other approaches described in the literature [10]. Though, it implies that the computationally projected rectangular 2D image is also represented as a rectangular picture in the real world. While this is indeed the case for almost all computer monitors, it can rarely be claimed for most front-projection based displays. In order to tackle this problem already in the process of 3D image generation, we extend our viewing and projection model such that corrections of the synthetically projected image are possible. Thus, physical projection errors due to projector misalignment in the real environment can be compensated for. This keystone correction can be accomplished as the final step in the transformation pipeline or, in OpenGL terminology, by initially loading a correction transformation matrix upon the projection matrix stack. This procedure is already taken care for with the statement `PostProjectionTransform();` in the pseudo code above.

Fig. 4 and Fig. 5 show the situation, in which the central projection line of the projector at position PS is not perfectly aligned with the normal vector n of the display screen (i.e. viewing window). In most practical situations it is possible to ensure that the image projector stays on-axis at least in the horizontal direction i.e. it is usually possible to assure that the upper and lower edge of the projection image be parallel as shown in Fig. 5. The question is, how the parameterisation of a correction transformation must be, given the real world position of a projector in relation to the projection screen?

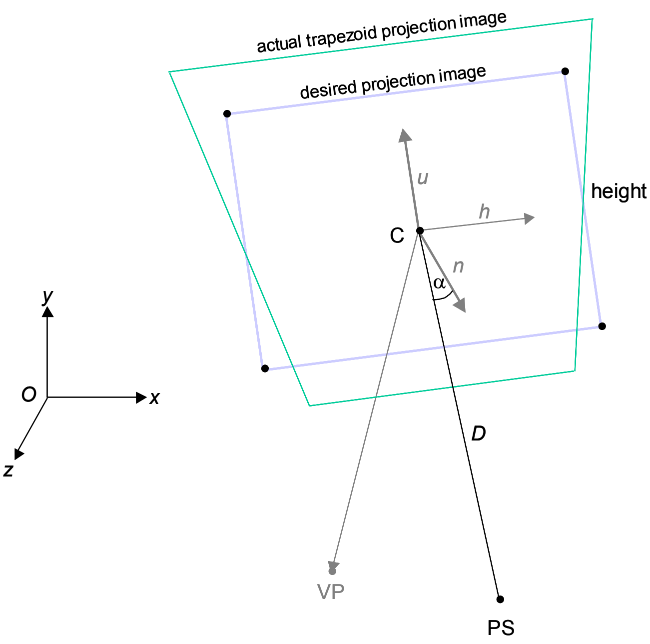

Fig. 5. The relationship between the general viewing metaphor for computational projection of virtual 3D objects and the trapezoid projection image generated by an off-axis positioned data projector.

Earlier work published by Sukthankar et al. [12] has been addressing this problem. The authors present a linear mapping function for image warping that is deducted from the comparison of an observed distorted picture with the desired keystone corrected picture. This approach does not make any assumption in regard to the spatial relationship between the projector and the projection screen. Instead it requires a parameterisation of the distorted 2D image. In our viewing model we intend to incorporate the actual spatial position of the projector within the physical environment into the calibration procedure. Thus, we derive the parameterisation of the correction matrix from knowledge about the spatial configuration of the image projector and the projection screen.

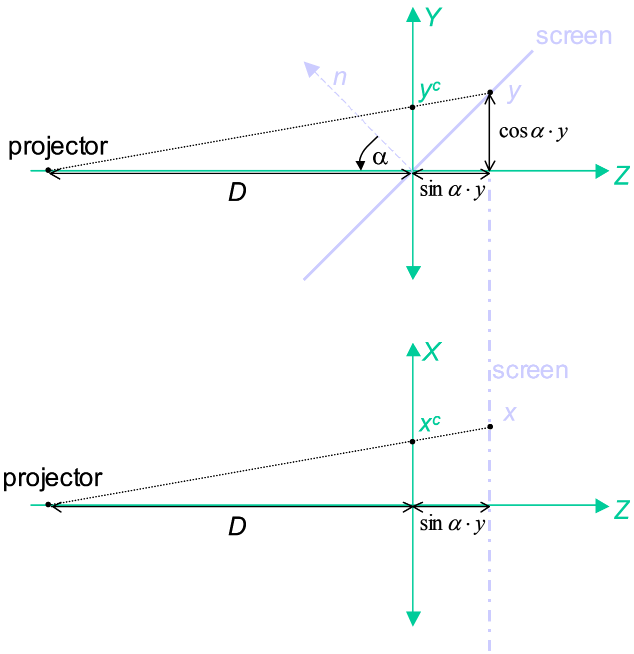

Fig. 6. The relationship between the projector local coordinate system of the projector and the coordinate system of the projection screen. We consider vertical off-axis projection that causes trapezoid distortions of the projected picture.

Fig. 6 depicts the local coordinate system of the projector. The central line of projection is vertically tilted off the axis of the screen coordinate system about the angle α. According to projection laws, the following relations hold true:

$$\frac{y^c}{\cos\alpha \cdot y} = \frac{D}{D + \sin\alpha \cdot y} \Rightarrow y^c = \frac{D \cdot \cos\alpha \cdot y}{D + \sin\alpha \cdot y} \tag{12}$$

and

$$\frac{x^c}{x} = \frac{D}{D + \sin\alpha \cdot y} \Rightarrow y^c = \frac{D \cdot x}{D + \sin\alpha \cdot y} \tag{13}$$

Where D is the distance between the projector unit and the point of intersection between the central projection line and the projection screen. The angle α is the vertical off-axis angle between the central line of the projection pyramid of the projector and the normal vector of the projection screen. Since all parameters in the viewing metaphor described until now are based on real world metrics, also D is measured in the same metrical units as the definition of all other parameters of the viewing and projection metaphor.

X and y are projected coordinates on-screen appearing in undistorted dimensions, and x^c and y^c are pre-distorted coordinates to be generated by the synthetic rendering pipeline to form a picture that is subsequently projected by the physical projector unit.
The formulas in (9) and (10) can be denoted in form of a homogenous transformation matrix such as to be pre-concatenated with the viewing matrix:

$$C = \begin{bmatrix} D & 0 & 0 & 0 \\ 0 & D \cdot \cos\alpha & 0 & \sin\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & D \end{bmatrix}$$

The procedure PostProjectionTransform() for keystone correction of the projection comprises therefore two steps: First, calculating C according to (14) and secondly multiplying the result with the current matrix on top of the projection matrix stack.

3.2 Adaptive texture re-projection on geometric objects

The interactive visualization of the mummy was based on the geometric objects that were subsequently acquired from the initial CT raw data. A list of the objects with their respective number of polygons is shown in Tab. 1. A graphical representation of the objects is given in Fig. 1. As can be seen, the visual quality of these polygon meshes is more than satisfactory. However, we found some drawbacks after trying to visualize those models interactively. One big obstacle was the large number of independent polygons, which caused frame rates clearly below 15 frames per second, even with state of the art graphics hardware. Another problem was the colour representation. The bones and also dried skin can be rendered with sufficient degree of realism by using appropriate material definitions. However, for the outer shell of the mummy, the use of the photo realistic textures was crucial in order to depict details of facial expression and woven linen wrappings. The model, which represents the outer layer of the mummy represents almost one third of the total number of polygons. In textured rendering mode, we encountered that significant performance increases were easier to accomplish by reducing moderately the number of textured polygons rather than decimating rigorously the number of non-textured polygons. While to our experience the texture performance in terms of fill rate never was a critical issue, we believe that expenses for texturing in the triangle set-up can explain the described phenomenon.

In order to find an optimally balanced configuration on various computer systems with different performances, we employed dynamic mesh simplification and decimation at program start up, to find an acceptable number of polygons for the textured part of the model. Our iterative mesh decimation algorithm utilises surface curvature parameters to identify and eliminate redundant polygons in the object. We will not expand the details of the algorithm in this place, because it is quite similar to approaches described earlier in the literature [13,14,15,16]. However, we address the question of how texture coordinates must be treated in the mesh simplification procedure in order to avoid excessive texture warping artefacts.

We tackle this problem utilising the fact that we have a-priori knowledge about how the textured object looks in the real world, and under which circumstances the photographic pictures where taken that form the texture image. We implemented a small interactive graphical editor program, which allowed us, starting from the initial photographic parameters, to define the position of one frontal and two lateral focal points of projections in relation to the virtual mummy object. Projecting the virtual mummy towards these focal points creates two-dimensional silhouettes of the mummy, which naturally should coincide with the respective photographic pictures represented in the texture image. The purpose of the editor program was to shift the 2D texture images and to interactively fine-tune the spatial relation between mummy, texture images and focal points such that the original photographic configuration was optimally reconstructed. In this configuration, a nearly perfect match between the silhouettes in the texture image and the contours of the projected virtual mummy object can be accomplished. This configuration was stored in an external file for later use.

Subsequently, whenever the mesh toplogy of the outer mummy layer has changed as a consequence of mesh simplification, the configuration data in the external file is reused.

A ray-tracing process is started that identifies intersection points within the texture image for lines between each vertex in the object and the appropriate focal projection point. In this way each vertex of the object is projected upon the texture image, and the corresponding 2D texture coordinate is written back into the vertex. This procedure is performed selectively for polygons depending on their surface orientation in relation to one of the three focal points.

Fig. 7 illustrates the spatial configuration between the texture images, focal points and the model of the external layer of the mummy. In the left picture, different colours on the mummy surface indicate that different focal points apply for respective surface points. The right side of Fig. 7 shows the result of the automatic texture coordinate generation. The advantage of the described procedure is that the chosen mesh decimation strategy can be optimised exclusively in terms of geometric optimisation criteria and does not need to be compromised by parametric constraints in texture and colour coordinate space.

3.3 Stroke based interaction with digitiser tablet

As was the intention with the development of this virtual mummy, interaction i.e. virtual dissection should be easy and intuitively to be used by non-expert users. A physically correct simulation of a dissection process does not really pose a problem. However, it would require advanced motion capture devices to recognise the user's gestures and body expression, and more importantly, due to its physical correctness it would demand much more trained perfection in the interaction behaviour of the user. We therefore decided to provide simpler tools for the user to peel off layer by layer and thus to uncover the inner structures of the mummy. This virtual peeler was technically accomplished by definition of a number of clipping planes that were associated with the different layers. We let the clipping planes be locked with the anatomical transversal direction, and allow the user to interactively manipulate the trans-axial positions of these planes.

To control these positions, we felt that a digitiser tablet was the most intuitive device to use. It features a pen as main input utensil and it allows affixing templates with self-explaining illustrations to the spot of the actual input.

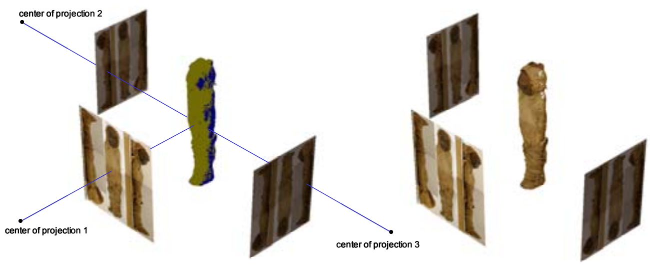

Fig. 7. An interactive editor allowed for fine-tuning of the spatial attitude between the virtual mummy and anticipated real world camera position, such that the sub-images coincide with the shape of the virtual mummy.

Input is based on strokes with the pen of the digitiser tablet. They are interpreted by the software and transformed into parameters to calibrate clipping planes or to activate other functions. Fig. 8 illustrates the graphical template affixed to the digitiser tablet. Peeling of the outer layer or the skin of the mummy is accomplished by striking along the central line of the respective mummy icon on the digitiser layout. Hereby, accurate input is not required. As long as the pen is guided in a certain distance to the central line, the software automatically projects the current position of the pen upon the central line and determines the proportion of the clipped area of the respective geometric object (see also top left area of Fig. 8).

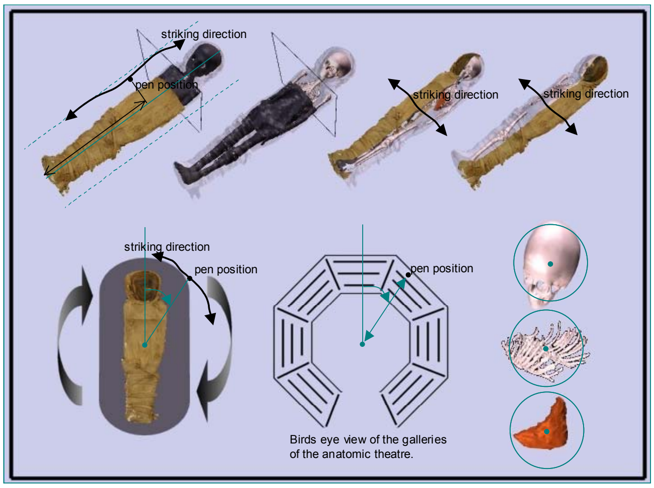

Fig. 8. Graphical template for digitiser tablet based interaction. The bold arrows indicate strokes performed by the user. Green lines are mathematical lines of reference.

Another way of slicing the mummy is to manipulate a clipping plane, which is simultaneously affecting rendering of both the linen lay and the skin layer of the mummy. The navigation of this clipping plane is similar to the previously described procedure, where the striking direction goes laterally to the mummy icon on the digitiser layout (top right icons on the digitiser table). In order to make the interaction more fetching, particle animation effects accompany the change of a clipping plane (see Fig. 9).
In addition, it is possible to horizontally turn the entire virtual mummy on top of the dissection table. This is due to the fact that the real dissection table, upon which the virtual mummy is recreated, can be turned. To change current turn angle, any point in the neighbourhood of the turntable icon on the digitiser may be hit. The program subsequently calculates the angle between the line from the hit point to the centroid of the icon and the vertical line of the digitiser template.
A similar modality is used to interactively adjust the observer's current viewing position in the galleries of the anatomic theatre. To this end, the balconies of the theatre are represented in an iconic picture from a birds eye view. After pointing into the respective area in this icon, the angle and distance to the centre of the anatomic theatre is retrieved by the program. The viewing position is directly fed into the projection pipeline of the application.
Finally, pointing to their corresponding icon in the digitiser layout activates an animation sequence for specific parts of the mummy. The animation lets anatomic structures (e.g. chest, liver and skull) pop up out of the mummy and enlarges them for further inspection. Here, the hit-point on the digitiser tablet must be situated within a specific circular area that contains the corresponding anatomic structure.

4. Results

The specific installation of the interactive virtual mummy in its historical context of the anatomic theatre can be seen in Fig. 9 (bottom picture). It shows the projection of the virtual imagery on a cotton cloth that covers the top of the dissection table. Even though the application has functionally been developed for dual-piped projection with dynamic head tracking in mind, it has for reasons of practicality been decided not to install specific motion tracking equipment. Instead, users may occupy any position in the gallery of the anatomic theatre, and by pointing on the digitiser layout chose their current viewing position. In that way, visitors can indeed also explore the significance of a correctly projected individual 3D image in comparison to the imagery generated for an alternate user with a different viewing position.
The bottom picture in Fig. 9 also shows the actual digitiser tablet with the graphical layout. The digitiser is connected to an extension cord that allows the visitors to pick up and take along the digitiser when walking up the gallery. An

information panel explains to the visitors in few concise statements how to use the digitiser-based interface for interaction.

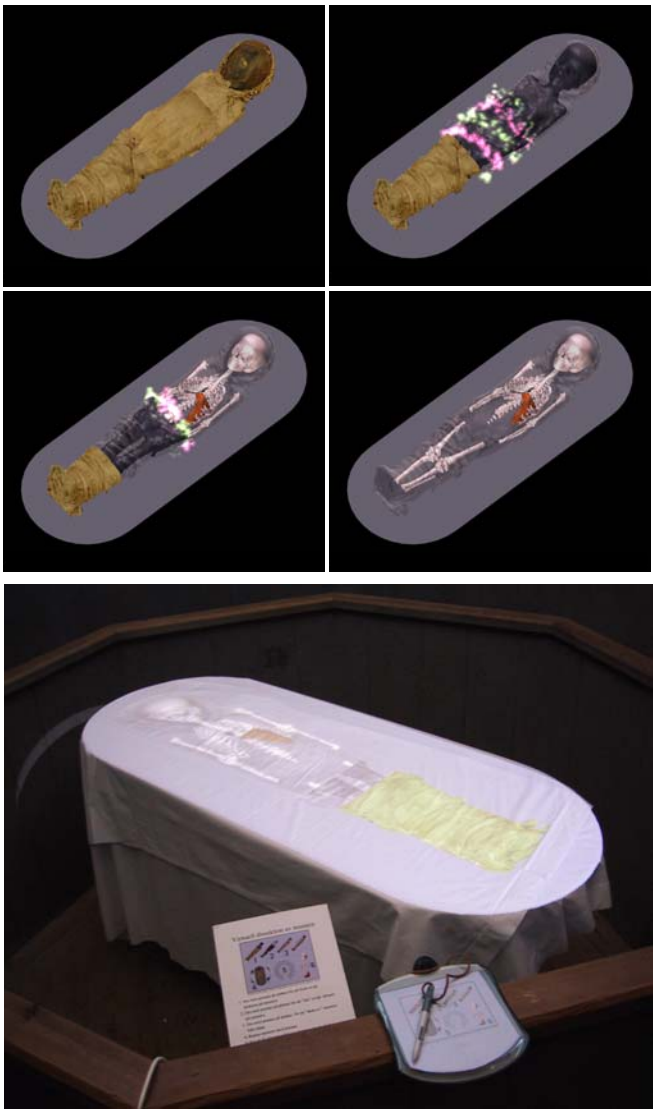

Fig. 9. Resulting graphical effects of the virtual mummy. Above, a sequence of successive peel-off of the mummy is illustrated. The bottom picture shows the actual installation in its historically authentic context.

In Fig. 9 upper part, the sequence of a typical interaction scenario is shown. For the purpose of clarity, the pictures have been projected and rendered for on screen presentation. As is evident, the different outer object layers of linen wrapping and subsequently the skin layer can be peeled of to uncover the interior constituents of the mummy. The dynamic motion of the clipping plane is supported by an animation of a colourful particle system. Runtime performance of the system is an important issue not only for real-time update of the observer's head position, which though was not enabled in the actual installation. But also for the animation of the particle system and for an immediate and smooth response in the interactive peeling procedure, high rendering speed is vital.

In the actual exhibition, a Pentium III based computer with 1200MHz clock frequency and 128 MB of RAM is used to render the simulation. It is equipped with a no-name graphics card that is based on GeForce2^{TM} technology, and which is equipped with 32 MB of on-board memory.

For rendering purpose we use the models listed in Tab.1. However, to accomplish higher frame rates, the textured model of the outer layer has been reduced by 19614 polygons. Therefore the total number of polygons for the mummy sums up to 104741 polygons. Fig. 9 exhibits the simplified geometric objects. In this model configuration, we could on average achieve a frame-rate of approximately 23 Hz.

5. Discussion

The keystone corrected viewing metaphor developed in this project has proved to be a powerful instrument to quickly set-up projection based VR applications in new environments that are not and cannot be adapted to the technical equipment commonly used in VR. The parameterisation of this viewing metaphor allows to model virtual objects and real world display components in the same conceptual modelling space. It allows simple on-site measurements of the projector's position, the viewer's location, and the situation of the projection screen to correctly calibrate the software with the real world situation. Being an intrinsic part of the projection pipeline, the keystone compensation function enabled us to overcome the excessive distortions that could not be handled by the projector's build-in compensation mechanisms. Modelling of equidistant regular patterns that were to appear on the projection surface, allowed us to verify the correct function of the algorithm on site. It should be noticed that in practice, the parameterisation of the projector's position usually must be adjusted by a vertical positional offset that is known and specific for the model of the projector. This is due to the fact that most available projectors today automatically create a correct rectangular projection image at a certain given limited off-axis position. This does however not invalidate the generalised keystone correction method according to (14) and as presented in Fig. 5 and 6.

With regard to the automatic generation of texture coordinates based on photographic re-projection of texture images we can state, that this method creates very satisfactory visual results and that it helped to overcome our rendering bottleneck. The method is generally applicable to many other application fields of image-based rendering. In combination with our mesh decimation algorithm, the results of the re-texturing procedure for various resolutions of the same object were more than satisfactory. This is mainly thanks to the fact that the mesh decimation procedure chosen does not introduce significant artefacts to the projection silhouette of the object. For extreme mesh simplifications that lead to evident deformation of object shape, however, the described texture re-mapping procedure potentially also leads to incorrect mappings. Whenever the projected silhouette of the object exceeds over the photographic representation in the texture image, texture elements from the texture background are erroneously mapped upon the object.

The means of object manipulation and input as described above provide an appropriate degree of freedom in interaction to the user. Satisfying our initial requirement, the virtual dissection is intuitively to understand and the carefully selected interaction paths do not confuse the users. As a proof of our interaction method we account, that during the opening ceremony of the exhibition the vice-chancellor of Uppsala University offhand performed an impeccable debut dissection of the virtual mummy in front of the spectators. Sporadic observations of visitors of the exhibition under a three-month period showed, that visitors in all ages and regardless of their gender quickly interacted with and performed a virtual dissection of the mummy on their own.

Despite the scepticism of the involved technical experts in this project, the effect of the installation in the authentic environment is enormous, even though dynamic head tracking and stereoscopic projection using dual pipe polarized light projections were not used as intended with the software.

6. Conclusion and future work

We have presented a computer graphics application in the cultural heritage of the ancient Egyptian mummies. It provides a powerful tool for non-expert users to interactively uncover internal features of the mummies that are otherwise difficult to appreciate. That, beyond this, the virtual dissection application has been installed in one of the few preserved historical anatomical theatres, excites an experience of a historic kind to the visitors of the museum.

The practical requirements of the installation environment necessitated various technically novel solutions. With regard to those achievements we conclude, that an easy to configure viewing metaphor for arbitrary viewer-to-screen relations with built-in keystone correction was developed. This generic method is vital for all projection based VR applications that have to cope with extreme off-axis projection conditions and arbitrary viewer and screen configurations.

Our continued research in this project will look into the networked aspects of this application in order to accomplish distributed collaborative experiences of the virtual mummy. We have engagements in the development of protocols for propagation of shared states in a distributed VR environment. They have been applied successfully in the context of distributed virtual learning environments and their application in the virtual mummy project would make this historic content available to a wider community. Other planned activities are to use parts of the virtual mummy as a platform to implement virtual excavations as an advanced learning tool in education of students in Archaeology. These developments are going to be part of the research in virtual learning environments as carried out with partners in the Wallenberg Global Learning Network.

7. Acknowledgements

The work presented is one of many efforts that have been made in the context of an exhibition on the cultural heritage of the ancient Egyptian mummies in Uppsala. The authors would wish to express their thanks to following people: Prof. Ewert Bengtsson, from the Center for Image Analysis, Uppsala University, who inspired us to this interesting project. His collaborator Magnus Kronnäs, who post-processed the raw data and who provided us with the polygonal models for the virtual mummy. Dr. Ing-Marie Munktell, director of the Museum Gustavianum, Uppsala University Museum and Geoffrey Metz, antiquary at Uppsala University Museum and postgraduate at the Institute of Egyptology at Uppsala University, for supporting us with digital photographs of the mummy and its historical background. Prof. Anders Magnusson, Department of Radiology, Uppsala University, for supplying the project with CT data acquired in the Uppsala Mummy Survey.

References

[1] Pickering R.B., Conces D.J. Jr., Braunstein E.M., Yurco F.
Three-dimensional computed tomography of the mummy Wenuhotep.
Am J Phys Anthropol, **83**(1), 49-55, 1990

[2] Baldock C., Hughes S.W., Whittaker D.K., Taylor J., Davis R., Spencer A.J., Tonge K., Sofat A.
3-D reconstruction of an ancient Egyptian mummy using X-ray computer tomography.
Journal of the Royal Society of Medicine, **87**(12), 806-808, 1994.

[3] Bou C., Pomar P., Pessey J.J., Rabino-Massa E.
Three-dimensional facial reconstruction of computerized tomography images by computer-aided design: example of an anthropologic study.
Rev Laryngol Otol Rhinol, **119**(5), 333-335, 1998.

[4] Recheis W., Weber G.W., Schafer K., Prossinger H., Knapp R., Seidler H., zur Nedden D.
New methods and techniques in anthropology.
Coll Antropol, **12**(2), 495-509, 1999

[5] Hodges L.F.
Time multiplexed stereoscopic computer

graphics.
IEEE Computer Graphics & Applications, **12**:20-30, 1992.

[6] Cruz-Neira C., Sandin D.J., DeFanti T.A. Surround-Screen Projected Virtual Reality: The Design and Implementation of the CAVE. Computer Graphics, **27**:135-142, 1993.

[7] Kenyon R.V., DeFanti T.A., Sandin D.J. Visual Requirements for Virtual Environment Generation. Journal of the Society for Information Display, **3** (4), 211-214, December, 1995.

[8] Krueger W., Bohn C., Fröhlich B., Scheuth H., Strauss W., Wesche G. The Responsive Workbench Environment. IEEE Computer Graphics and Applications, **14**(3), 12-15, 1994.

[9] Seipel S. Design of a 3D workbench interface for training in dental implantology. In : Cesnik B. et al (Eds) : MEDINFO98, IOS Press, Amsterdam, 907-911, 1998.

[10] Carrozzo M., Lacquaniti F. Geometric transformations for displaying virtual objects on stereoscopic devices. Computer & Graphics, **21**(3), 329-334, 1997.

[11] Robinett W., Holloway R. The visual display transformation for virtual reality. Presence, **4**:1-23, 1995.

[12] Sukthankar R., Stockton R., Mullin M. Automatic Keystone Correction for Camera-assisted Presentation Interfaces. Proceedings of International Conference on Multimedia Interfaces, October, 2000.

[13] Schroeder W.J., Zarge J.A., Lorensen W.E. Decimation of Triangle Meshes. In Proc. ACM SIGGRAPH '92, pp 65-69.

[14] Hoppe H. Progressive meshes. In Proc. ACM SIGGRAPH '96, pp 99-108.

[15] Kobbelt L., Campagna S., Vorsatz J., Seidel H.-P. Interactive Multi-Resolution Modeling on Arbitrary Meshes. In Proc. ACM SIGGRAPH '98, pp. 105-114.

[16] Lau R.W.H., Green M., To D., Wong J. Real-Time Continuous Multi-Resolution Method for Models of Arbitrary Topology. Presence: Teleoperators and Virtual Environments, MIT Press, **7**(1), 22-35, 1998

Recent technical reports from the Department of Information Technology

2001-012 Friedhelm Stappert, Andreas Ermedahl and Jakob Engblom: *Efficient Longest Executable Path Search for Programs with Complex Flows and Pipeline Effects*

2001-013 Eva Olsson, Lena Kecklund, Michael Ingre and Anders Jansson: *Lokförarens informationsmiljö och ATC. Ett användarperspektiv*

2001-014 Paul Pettersson and Sergio Yovine: *Workshop on Real-Time Tools (Proceedings)*

2001-015 Magnus Svärd: *On coordinate transformations for summation-by-parts operators*

2001-016 Robert Stjernström: *User-Centred Design of a Train Driver Display*

2001-017 Henrik Björklund, Viktor Petersson and Sergei Vorobyov: *Experiments with Iterative Improvement Algorithms on Completely Unimodal Hypercubes*

2001-018 Emad Abd-Elrady: *An adaptive grid point RPEM algorithm for harmonic signal modeling*

2001-019 Wendy Kress and Jonas Nilsson: *Boundary conditions and estimates for the linearized Navier-Stokes equations on staggered grids*

2001-020 Emmanuel Beffara and Sergei Vorobyov: *Adapting Gurvich-Karzanov-Khachiyan's Algorithm for Parity Games: Implementation and Experimentation*

2001-021 Inger Boivie: *Usability and Design Decisions in Software Development*

2001-022 Pierre Flener, Alan Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel, Justin Pearson and Toby Walsh: *Symmetry in Matrix Models*

2001-023 Pierre Flener, Alan Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel and Toby Walsh: *Matrix Modelling*

2001-024 Larisa Beilina, Klas Samuelsson and Krister Åhlander: *A hybrid method for the wave equation*

2001-025 Emmanuel Beffara and Sergei Vorobyov: *Is Randomized Gurvich-Karzanov-Khachiyan's Algorithm for Parity Games Polynomial?*

2001-026 Jan Gulliksen and Inger Boivie: *Usability Throughout the Entire Software Development Lifecycle - A Summary of the INTERACT 2001 Workshop*

2001-027 K. Mahata, T. Söderstr öm, M. Mossberg, L. Hillstr öm and S. Mousavi: *On the use of flexural wave propagation experiments for identification of complex modulus*

2001-028 Magnus Berggren: *Wireless communication in telemedicine using Bluetooth and IEEE 802.11b*

2001-029 Henrik Lundgren, David Lundberg, Johan Nielsen, Erik Nordström and Christian Tschudin: *A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations*

2001-030 Jakob Engblom, Andreas Ermedahl and Friedhelm Stappert: *Validating a Worst-Case Execution Time Analysis Method for an Embedded Processor*

2002-001 Sônia M. Gomes and Bertil Gustafsson: *Combining Wavelets with Finite Differences: Consistency Analysis*

2002-002 Stefan Seipel and Mikael Lindkvist: *Methods and application of interactive 3D computer graphics in anthropology*

UPPSALA
UNIVERSITET

January 2002
ISSN 1404-3203
http://www.it.uu.se/