

Visualizing Archaeological Excavations based on Unity3D

Thomas Trautner*

Supervised by: Gerd Hesina†

VRVis Research Center, Vienna / Austria

Institute of Computer Graphics - Vienna University of Technology, Vienna / Austria

Abstract

As part of an archaeological excavation, huge amounts of different types of data, for example laser scan point-clouds, triangulated surface meshes, pictures or drawings of finds, find attributes like location, age, condition and description or layers of excavated earth are collected. This detailed documentation is important to give archaeologists the possibility to analyze the collected data at a later date, since the find spot might not be accessible anymore. Unfortunately, all the accumulated data is separately saved and consequently complex to explore.

Therefore, we present a novel solution that allows the user to digitally explore a virtual archaeological excavation in real-time. With our approach, we can not only visualize different types of textured meshes and finds, but also allow the user to draw on surfaces to mark areas of certain interest that need further exploration, enable explosion views to investigate composition of different layers of earth and arbitrary slicing of the three-dimensional mesh structure to better visualize cross-sections, and an easier tracing of accumulation points of finds. The result of this work is a new powerful tool that will support the analysis of future excavations. All results and the implementation itself will be presented as part of this work.

Keywords: Unity3D, Archaeology, Visualization

1 Introduction

This work complements the already implemented *Harris Matrix Composer* [a], abbreviated below as *HMC*. The HMC is an important tool archaeologists use for the documentation of archaeological sites. Our 3D viewer is an additional component of the new *Harris Matrix Composer - Plus* system. The viewer is a real-time renderer with a GUI shown as a toolbox which allows the user to further explore and verify a given three-dimensional data set. This allows the user to visualize selections and/or combinations of all types of stratigraphic units within the HMC.

The HMC is based on the *Harris Matrix* which was invented in 1975 by Edward Harris [8]. During an excavation, this matrix is used to document the stratigraphic

relations of dug out sediments. Every unit of stratification – for example remains of wood or brick walls, a basement, or inclusions – is displayed as a single node in the hierarchical graph. Figure 1 shows a typical Harris Matrix.

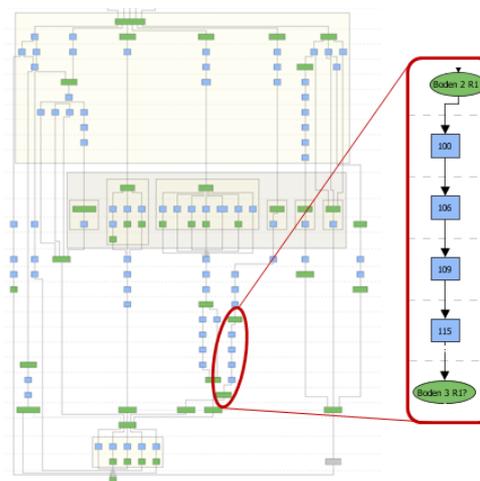


Figure 1: Example of a Harris Matrix within the Harris Matrix Composer. It is a two-dimensional and graph-like representation of dug out stratigraphical layers. Additionally we highlight a subdivision which represents room 1 of the excavated ruin in Falkenstein.

A typical HMC matrix starts with a circular green unit at the top. It represents the top surface of the archaeological excavation, for example measured by a laser-scan. The lowest unit is as well a circular green unit, which stands for the lowest excavated stratigraphic layer. Every other unit in-between consists of at least a top- and a bottom-surface, or a hull of these two surfaces. Additionally, a HMC matrix enables an accurate assignment of the find locations. Every path visualizes a different location, and therefore a subdivision of the excavated area can be easily identified.

Unfortunately, beside all the already mentioned advantages of a two-dimensional graph-like representation, domain experts from the field of archeology (like the *Ludwig Boltzmann Institut für Archäologische Prospektion und Virtuelle Archäologie* [c]) came to the conclusion that it is necessary to explore not only a graph-like representation, but also the real excavated three-dimensional data set. This is essential to better understand an excavation and easily track possible mistakes made during the docu-

*thomas.trautner@tuwien.ac.at

†hesina@vrvis.at

mentation process – like, for example, wrong classification or age determination of layers.

Therefore we extended the HMC and implemented this 3D viewer which allows archaeologists to (1) compare a *Harris Matrix* with measured three-dimensional data (2) perform manipulations on the data in real time and further (3) verify assumptions and findings. Doing this without our 3D viewer is extremely difficult and time-consuming, because all layers of earth are irretrievably removed during an excavation.

2 Related Work

A very similar approach of visualizing archaeological excavations is presented by J. Cosmas et al. [4]. The authors propose a tool to visualize building parts, finds, stratigraphical layers and textures in situ, and furthermore provide advanced 3D reconstruction techniques. This is done by storing all different types of data in a multimedia database. The result is used for presentation or publication purposes.

Another approach presents a virtual reality tool [2] which allows the user to measure stratigraphical units like curvature, length, thickness, height or volume to enhance archaeological fieldwork. Furthermore it allows the user to calculate volumes and simulate archaeological formations and deformation processes which is essential for understanding an excavation.

The work of Vote [13] focuses on visualizing the Great Temple of Petra based on post-excavation archaeological analysis. As part of this work they analyzed four different prototypes. First of all a conceptual model that automatically assigns chronology by using finds and other datable objects, secondly a three-dimensional database and a *Geographical Information System* (GIS) visualization software, thirdly visualizations using a virtual-reality interface and finally they focused on improving visual perception using lighting and coloration.

Dellepiane et al. [5] present a technique of interactive slicing, where a slicing plane is moved further away toward the viewing direction to visualize disparity between two different time steps – a technique that is also very useful for archaeological visualizations.

Benko et al. [3] focus on a mixed-reality approach implemented for multiple users. It supports the tracking of see-through head mounted displays and multi-touch table surfaces. The final visualization is similar to our approach based on the Harris Matrix.

Additionally, we considered a summary [9] of five papers that propose different approaches to visualize seismic measurements. This is done to allow the user to further explore and analyze possible oil or gas reserves. For optimal use the author recommends the following expressive visualization and rapid interaction techniques: the time of creation of such an illustration must be reduced, the program must independently recognize important stratigraph-

ical layers, and the exploration should be as simple as possible for the user.

Remondino and Campana [10] present image-based approaches to capture detailed 3D information of archaeological excavations. This is done with cost-efficient approaches like taking pictures of finds from different angles, and finally computing the three-dimensional structure. Therefore we implemented a run-time object loader for our HMC-Plus 3D viewer which is able to load and visualize such models as well.

A similar low-cost approach is presented by Doneus et al. [6]. Instead of expensive techniques like measurements with laser scanners, they use a technology called *Structure from Motion* [12] and create a three-dimensional point cloud. This approach requires only minimal technical knowledge and user interaction, and is therefore straightforward to use.

Allen et al. [1] introduce a 3D Modeling Pipeline for archaeological excavations and finds. First, the excavation site is either scanned by a laser scanner or pictured with cameras. In the next step, the site is represented as 3D model, and additional context like background images, videos or GIS data is added.

Santos et al. [11] focus on realistic rendering of archaeological excavations. They concentrate on illumination methods like global illumination. The paper presents different approaches that guarantee a frame-rate of at least 10 images per second and the possibility to change the view point dynamically. Since lighting is such an important factor for a realistic perception of a scene, we allow the user to dynamically place light sources in the scene with our HMC-Plus 3D viewer.

To provide stereoscopic rendering we further implemented a technique called off-axis rendering, which was described by Grasberger [7]. His work presents different approaches of stereo rendering and explains in detail their use, the advantages and disadvantages.

3 Implementation

Our main goal was building a tool that allows the user to scientifically visualize archaeological excavations. The resulting 3D viewer should be dynamic, customizable, easily extendable and has to support the import of high-resolution geometry and very large geometric models. Therefore, we chose the Unity3D Game Engine [b] as an optimal framework for our implementation. Furthermore, Unity 3D offers the possibility to build a 3D viewer for different platforms.

3.1 Object Import and Data Preparation

We wanted our rendering framework to be as independent and compact as possible. Therefore, we implemented a 3D viewer and partially extended it with already existing libraries that were well-suited for our approach – like for

example an improved file browser dialog [d] that allows us to display different folder structures including all their files during runtime. By additionally applying a filter function that allows the user to select only files with an (*.obj) extension, we managed to optimize the complex handling of the numerous 3D files. We furthermore expect the following naming restrictions to guarantee a successful mesh import:

```
[0-9] - [000-350] tbs_ [0-99] - [A-Za-z0-9]? .obj
[Sub-Mesh] - [Name] - [Room] - [Information].obj
```

Example:

```
0_000tbs_0_falkenstein_smart.obj
0_000tbs_1_falkenstein_smart.obj
```

Sub-Mesh: We use this index to detect if a mesh consists of smaller sub-meshes. In the given example the two objects are named *000tbs* and belong to the same sub-mesh with index 0. If one of them is selected by the user during a load object operation, the other one will automatically be loaded as well.

Name: This name is displayed if the mouse cursor is moved over the mesh in our rendering window. The name consists of three digits that represent the layer and the acronym *tbs* which stands for *top-bottom surface*. The lowest layer of excavation should always be called *000tbs*. All other layers can be called according to their HMC-label, although the highest possible number is currently *350tbs*. This limitation results from the current maximum number of tags that are assigned dynamically during runtime. Unfortunately, tags must be hard-coded and cannot be created depending on variables which are passed during run-time.

Room: The room number is important to enable future versions of the HMC-Plus 3D viewer to perform manipulations only on individual rooms of the currently selected mesh (for example only explode layers of a single room). The current version of the 3D viewer explodes all layers equally independent from the selected mesh and room it belongs to.

Information: This part is not required by the 3D viewer. It can be used to store additional information assigned by the user. If there is no need for further information, it can be left out.

Unfortunately, our external triangulation library [e] and Unity3D itself have a maximum vertex count limit of 65.534 vertices per mesh. If such a big mesh is imported into the editor, Unity3D will automatically divide it into default sub-meshes. After building the project, or during runtime, this additional processing step is not possible anymore. Therefore, we expect the user to either load smaller meshes or divide meshes which have more vertices into sub-meshes manually in a preprocessing step.

3.2 Mesh Selection

If a layer is selected by clicking the left mouse button, the outline of the selected object is changed to red, which can be seen in Figure 2.

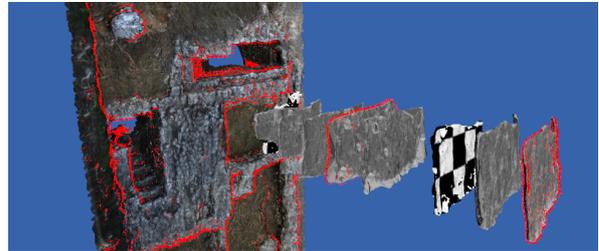


Figure 2: An example scene where three different meshes were selected. Their shader is changed from a diffuse shader to a red outline-shader. This feature allows the user to easily see which layers are currently selected.

As we can see the mesh is still textured, but the shader is changed from a simple diffuse shader to an outline-shader with red as its main color. If the "Ctrl" button is pressed and another object is selected, this object is selected as well, and their shader is changed to an outline shader. In Figure 2, three different layers were selected: First of all the *000tbs* ground layer, then layer *074tbs* in the middle, and the highest and therefore latest layer *006tbs* on the rightmost side of the screen. If the "Ctrl" button is not clicked, the first selected object will be deselected after another element is selected. By clicking on the background all currently selected meshes will be deselected.

Especially this feature will be essential for future versions of the HMC-Plus 3D viewer. It will allow the user to select stratigraphical layers within the 3D viewer and directly get their position in the two-dimensional Harris Matrix.

3.2.1 Functioning of the Outline Shader

The result of the dot product is a scalar which represents the cosine of the angle between two vectors. If the dot product of the current viewing vector and the surface normal is zero, we simply change the pixel color to the outline color. If the dot product is not zero, we do a texture look up for the current pixel and calculate the dot product of the surface normal and the light direction to illuminate the pixel with a diffuse illumination model. An example of this algorithm can be seen in Figure 3.

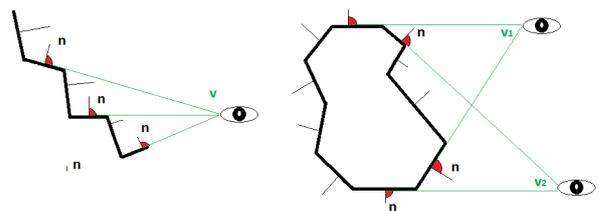


Figure 3: Low resolution meshes with an outline-shader

In the case demonstrated in Figure 3, the black polygon line represents an example layer which has an outline shader attached to it. Every triangle of this mesh has a surface normal which is represented by the vectors called n . The current camera position is represented by the eye with the corresponding viewing vectors called v . If the dot product is zero, the surface color will be changed to red, which is represented by the red angle in the middle. In this case vector n is orthogonal to vector v .

In Figure 3, we visualize two special cases: the cross-section of a triangulated surface without a correct red outline and the cross-section of a concave polygon with two viewpoints from which a correct outline is drawn.

3.3 Show Finds

During the excavation process, usually different items or materials like coins, glass, wood, porcelain, iron or others are found. To provide interactive exploration of find locations, our 3D viewer is capable of visualizing them as well. An example of this can be seen in Figure 4.



Figure 4: Example of finds of an excavation. They are represented as colored boxes whereas the color depends on the material properties of the find. Furthermore an additional description is shown if the mouse is moved over a box (for example year dates, the material, or layer number in which it was found).

Finds are usually represented by a point-cloud file which stores all X-, Y- and Z-coordinates (*.obj) and an additional description file (*.txt), which includes the layer number where it was found, the material and an additional description for every find. As soon as the 3D viewer has imported all finds, colored boxes will be displayed. Every box represents a single find.

Currently the color is chosen randomly, but finds with similar materials are always colored the same: For example, every silver coin is colored red, and every copper coin blue. To guarantee that the colors are significantly different and the same color is not selected twice, we used a hash-function that uses all the available find-information as seed to calculate the color. To improve the comparison of finds across different excavations, the next version of our 3D viewer will allow the user to select certain color maps to be able to use the same colors for materials which were already used in previous models of archaeological sites and only define new colors for new materials.

Furthermore, the user can change the color distribution, so that all finds of a certain layer will get the same color.

This feature allows the user to easily see which finds were found in which layer and where accumulations may be.

3.4 Slice Surface-Layers

Stratigraphy is a meaningful subsection of archaeology. Therefore it is important for archaeologists to understand how layers have moved, changed and sedimented over the years. Knowing this allows for example the age determination of finds. Unfortunately layers are often not well separated and therefore the Harris Matrix is not expressive enough. Therefore we implemented another important feature which allows the user to explore the structure of all stratigraphical layers of an excavation in explosion view. After the user has selected a layer, the 3D viewer will automatically calculate the bounding box of the selected mesh, and a slicing plane is shown. This is depicted in Figure 5.

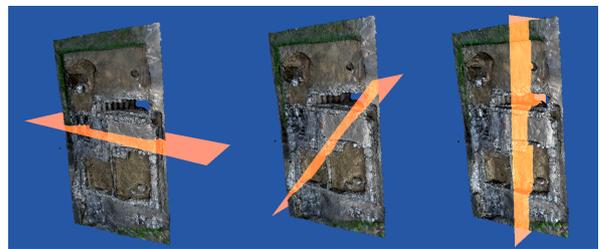


Figure 5: Different positions of the slicing plane

The dimension of the slicing plane depends on the dimensions of the bounding box. To guarantee that the slicing plane is always large enough to cut through the whole currently selected layer, the greatest dimension of the bounding box determines the width of the slicing plane. The user can then change the position and orientation of the slicing plane. In some special cases, like for example a diagonal position of the slicing plane, the greatest dimension of the bounding box is not enough. Therefore, we will increase the width and use the greatest distance between two points of the bounding box in the next version of the 3D viewer.

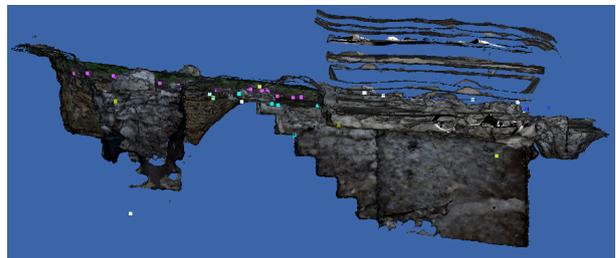


Figure 6: In this Figure we present a possible slicing view of an excavation. To visualize cross-sections of surfaces our 3D viewer uses orthographic cameras. Furthermore it is still possible to visualize finds and enable the explosion mode.

Additionally, we want the user to be able to place the plane exactly at the desired location. To simplify this task,

the slicing plane is semitransparent, and its color differs significantly from typical soil colors to further strengthen the contrast. The slicing view uses two orthographic cameras to visualize cross-sections of surfaces. An example of the sliced model can be seen in Figure 6. All the finds are visible and room 1 is currently in explosion mode.

3.5 Explosion View

Another important feature we implemented allows the user to explore the structure of all stratigraphical layers of an excavation in explosion view. Using this type of visualization, the hierarchical order and composition of all layers are better visible and therefore easier to analyze. The resulting animation steps can be seen in Figure 7.

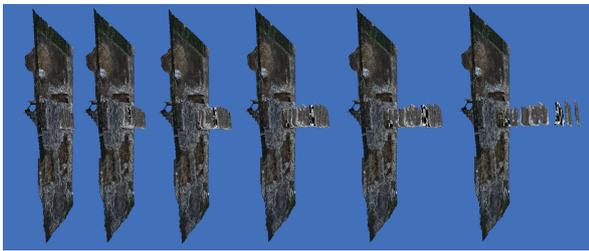


Figure 7: Explosion view animation sequence

Figure 8 shows a graphical explanation of our algorithm. The *expander* variable is a float value which controls the factor of explosion. Depending on the total number of layers called *k*, it is multiplied with *k* minus the current layer index.

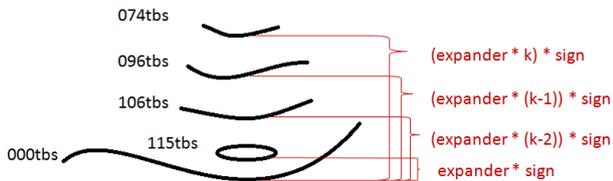


Figure 8: Simplified explosion view explanation

Therefore the collection of expandable layers is always sorted in descending order. The sign factor depends on the direction in which the mouse wheel was turned. With this adaption the same algorithm can either perform an explosion to enlarge the offset or an inverse explosion and move all layers back to their original position.

3.6 Brushing

The brushing tool can be used to mark certain regions of interest. This is important to identify future excavation sites, potential errors, overlooked finds and further to be able to easily export and exchange this information globally. To enable this, we implemented multilayer-texturing using an additional alpha texture. If our 3D viewer imports a mesh it will have two texture layers. The first layer

is the texture that the material file was pointing at. The second layer will be added by the viewer. It is a simple alpha-texture that is used for all the brushing operations. After selecting a color and enabling the brushing tool, the user can use the left mouse button to draw on a surface and the right button to erase already made drawings. To remove drawings, they are changed back to the original alpha value. The brush size is calculated dynamically, but currently only quadratic kernels are supported. Another characteristic can be observed in Figure 9: If the brushing is performed repeatedly on the same spot, the opacity of the color increases.

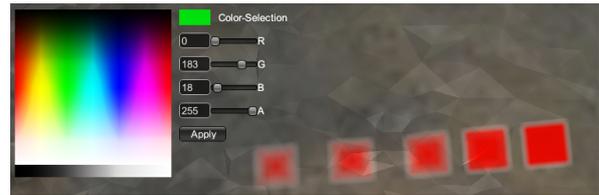


Figure 9: Color selection and kernel with different opacities

In the top left corner of Figure 9 the color selection [f] is visible. It can be used to select RGBA colors either by simply clicking on a color, or by inserting values from 0 to 255 for every red, green, blue and alpha channel.

After the brushing operation the alpha texture can be exported and for example imported the next time the users opens his model. It will be saved as (*.png) in the folder for textures and screenshots of the opened scene. In addition, the date and time is recorded in the name of the file. For example:

6-18-2014_10-42-50_AM_Texture_44

[month]-[day]-[year]-[hour]-[min]-[sec]-[AM/PM]-
Texture_[sequence number].png

An example of how marked regions could look like can be seen in Figure 10.



Figure 10: Example of marked regions

3.7 Adding Light Sources

To enhance the perception of the structure of the reconstructed geometry, the scene can be illuminated manually during runtime. The current version of the 3D viewer supports point lights; however, future versions will include

different types of light sources like for example spot-lights or directional-lights. Once a light source is added to the scene, its intensity can be changed. An example of how different intensities could look like is shown in Figure 11.

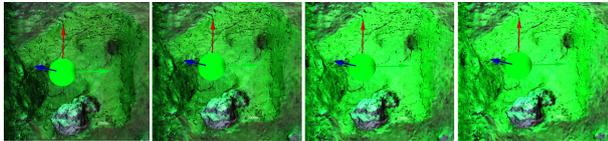


Figure 11: Controlling the intensity of a point-light source using the mouse-wheel

To manipulate the light source position three arrows representing the X-, Y- and Z-axis are displayed. By clicking on one of them with either the left or right mouse button, the position can be moved up or down along this axis respectively.

3.8 Name Highlighting

If the mouse cursor moves over a mesh, we automatically display the name of the mesh. We do this by continuously emitting rays from the mouse position into the viewing-direction. If such a ray hits a mesh-collider, we print a label with the corresponding mesh name next to the cursor. Thereby we allow the user to interactively explore the given data set without having to look up every layer within the Harris Matrix.

4 Additional GUI-Settings

Apart from the already mentioned exploration features, we allow the user to change some basic settings of the rendering system as well. These adjustments include the actual type of camera and the used render mode.

4.1 Camera modes

Currently we support two different camera modes: The first mode, which can be seen in Figure 12, is called *Fly-Through-Mode*. The second mode, which is visible in Figure 13, is called *Origin-Rotation-Mode*.

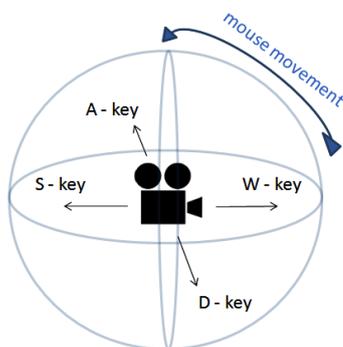


Figure 12: Fly-Through-Mode explanation

4.1.1 Fly-Through-Mode:

The viewing direction can be changed by moving the mouse, and the camera position can be changed by using the W, A, S and D keys.

4.1.2 Origin-Rotation-Mode:

Using this mode the camera moves circularly around its center-point. The distance to the center-point can be changed by using the mouse-wheel.

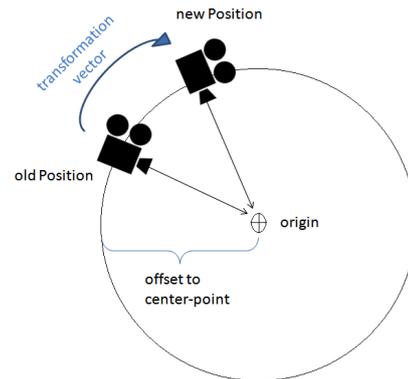


Figure 13: Origin-Rotation-Mode explanation

Additionally, this mode consists of two cameras to provide stereoscopic rendering. To render these images, we implemented off-axis-projection by using different projection matrices.

5 Results

The most important performance criteria for us was the applicability in real-time to allow the user to interactively explore our visualization. Therefore we focused mainly on GPU and CPU consumption and the average number of frames per second in this section. The following hardware settings were used on the testing computer:

Operating system: 64-bit Windows 7 Home
RAM: 16 GB
CPU: Intel(R) Core(TM) i7 -3930K CPU @ 3.20Ghz
Graphic card: NVIDIA GeForce GTX 690 with 4096 MB memory

The scene we used for testing consists of twelve different stratigraphic layers. Every layer is a triangulated and textured high-resolution point-cloud (*.obj). The biggest layer we used is the 000tbs ground layer which consists of 314.347 vertices and has an overall size of 20,17 MB. Furthermore, we used a 1024x1024 picture (*.png) taken during the excavation to texture the ground layer and a 128x128 checkerboard pattern (*.png) to texture all other sub-meshes. An overview of all object and texture sizes can be seen in Table 1.

| Number of layers | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Added layer | 000tbs | 006tbs | 020tbs | 032tbs | 035tbs | 054tbs | 074tbs | 096tbs | 100tbs | 106tbs | 109tbs | 115tbs |
| Vertices in layer | 314347 | 6070 | 10931 | 10543 | 7734 | 10589 | 10393 | 8793 | 3067 | 10403 | 5731 | 3465 |
| Vertices in scene | 314347 | 320417 | 331348 | 341891 | 349625 | 360214 | 370607 | 379400 | 382467 | 392870 | 398601 | 402066 |
| Size of *.obj file(s) (kB) | 20.170 | 329 | 618 | 800 | 706 | 616 | 612 | 519 | 162 | 593 | 395 | 311 |
| Texturesize of layer (kB) | 2.370 | 4,74 | 4,74 | 4,74 | 4,74 | 4,74 | 4,74 | 4,74 | 4,74 | 4,74 | 4,74 | 4,74 |

Table 1: Overview of the example scene *Falkenstein*

| Number of layers | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|------------------|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Added layer | - | 000tbs | 006tbs | 020tbs | 032tbs | 035tbs | 054tbs | 074tbs | 096tbs | 100tbs | 106tbs | 109tbs | 115tbs |
| lowest FPS | 140 | 133 | 120 | 103 | 88 | 80 | 73 | 69 | 43 | 35 | 31 | 22 | 19 |
| highest FPS | 173 | 168 | 165 | 152 | 149 | 148 | 132 | 113 | 109 | 91 | 80 | 75 | 60 |
| average FPS | 156 | 148 | 142 | 135 | 134 | 125 | 115 | 95 | 73 | 48 | 45 | 41 | 40 |

Table 2: Overview of the lowest, highest and average frame rate

Usually, an average frame rate of at least 35 frames per second is needed to ensure interactivity in real-time. Table 2 shows a minimum average frame rate of 40 FPS. The lowest frame rate occurs usually during and immediately after the importing of a new layer. Thereby we can guarantee that our viewer is able to deal with large data sets with a high vertex count.

6 Conclusions

The HMC-Plus 3D viewer was already presented to the potential end-user group of archaeologists of the *Ludwig Boltzmann Institut für Archäologische Prospektion und Virtuelle Archäologie* [c]. They found the 3D viewer to be especially helpful and easy to use – on one hand to watch cross-sections and explosion views to analyze structures and their relations, and on the other hand the stereo rendering for public presentations such as press conferences.

Although we built a stand-alone viewer and could therefore not totally benefit of a fully developed game engine like in our case Unity3D, we were able to achieve excellent results. We presented the development of a powerful tool, which is designed to simplify the future work of archaeologists. With this solution, it is possible to not only analyze a graph-representation of an archaeological excavation, but visually represent all measured data at run time. This provides a virtual exploration for archaeologists who were not able to physically visit the excavation site. In addition, it allows archaeologists to research different stratigraphic layers at the same time. This is usually difficult, as younger layers, which are mostly located above older layers, must be irretrievably removed to reach lower surfaces. With our approach no information is lost, and even younger surface layers can be studied easily.

7 Future Work

The goal of our future work will be to develop an interface between this viewer and other parts of the Harris Matrix Composer - Plus system. It will then be possible to not only investigate the three-dimensional data set, but also instantly determine its position in the Harris Matrix and vice versa. As result, we want to create a single, even more powerful tool out of already developed components, which can be seen in Figure 14.

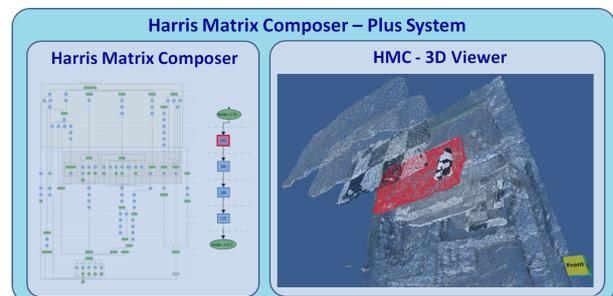


Figure 14: The final Harris Matrix Composer - Plus System will consist out of already implemented components for example the HMC and the 3D viewer. If a unit in the HMC is selected it will be automatically highlighted in the 3D viewer.

We further plan to replace the box-placeholder of the finds with their exact 3D representations. In addition we want to apply the Explosion mode not only on the stratigraphical layers but also to the finds, develop other intuitive camera modes, and implement ambient occlusion to improve the spatial perception for the users. Furthermore we plan to implement selection based manipulations like exploded views which only affect certain areas or rooms of an excavation.

Finally we want to expand our user study to ensure that our HMC-Plus system supports most of the archaeological work and techniques.

8 Acknowledgement

At this point I want to thank Michael Schwärzler from the *VRVis Research Center* for reviewing and always suggesting helpful improvement.

Furthermore we want to thank *Ludwig Boltzmann Institut für Archäologische Prospektion und Virtuelle Archäologie [c]* which generously provided us with actual data from their excavation in *Falkenstein*.

This work has been supported by a grant from the *Austrian Science Fund (FWF): P24597-N23 (VISAR)*.

References

- [1] Peter K. Allen, Steven K. Feiner, Alejandro Troccoli, Hrvoje Benko, Edward Ishak, and Benjamin Smith. Seeing into the past: Creating a 3d modeling pipeline for archaeological visualization. *Proceedings of the 2nd International Symposium on 3D Data Processing*, 2004.
- [2] Juan A. Barcelo, Oscar de Castro, David Travet, and Oriol Vicente. A 3d model of an archaeological excavation. *The Digital Heritage of Archaeology*, Computer Applications and Quantitative methods in Archaeology, 2003.
- [3] Hrvoje Benko, Edward W. Ishak, and Steven Feiner. Collaborative mixed reality visualization of an archaeological excavation. *Workshop on Collaborative Virtual Reality and Visualization (CVRV)*, 2003.
- [4] John Cosmas, Take Itegaki, Damain Green, Edward Grabczewski, Fred Weimer, Luc Van Gool, Alexy Zalesny, Desi Vanrintel, Franz Leberl, Markus Grabner, Konrad Schindler, Konrad Karner, Michael Gervautz, Stefan Hynst, Marc Waelkens, Marc Pollefeys, Roland DeGeest, Robert Sablatnig, and Martin Kampel. 3d murale: A multimedia system for archaeology. *Proceedings of the 2001 conference on Virtual Reality, archaeology, and cultural heritage*, pages 297–306, 2001.
- [5] Matteo Dellepiane, Nicolo DellUnto, Marco Callieri, Stefan Lindgren, and Roberto Scopigno. Archeological excavation monitoring using dense stereo matching techniques. *Journal of Cultural Heritage*, 2012.
- [6] Michael Doneus, Geert Verhoeven, Martin Fera, Christian Briese, Matthias Kucera, and Wolfgang Neubauer. From deposit to point cloud - a study of low-cost computer vision approaches for the straightforward documentation of archaeological excavation. *Geoinformatics CTU FCE 2011*, 2011.
- [7] Herbert Grasberger. Introduction to stereo rendering. *Student Project, Institute of Computer Graphics and Algorithms - Vienna University of Technology*, 2008.
- [8] Edward C. Harris. The stratigraphic sequence: A question of time. *World Archaeology*, 7, 1975.
- [9] Daniel Patel. Expressive visualization and rapid interpretation of seismic volumes. *Thesis for the degree of Philosophiae Doctor (PhD) at the University of Bergen - Norway*, 2009.
- [10] Fabio Remondino and Stefano Campana. Fast and detailed digital documentation of archaeological excavations and heritage artifacts. *Proceedings of the 35th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA)*, 2008.
- [11] Luis Paulo Santos, Vitor Coelho, Paulo Bernardes, and Alberto Proenca. High fidelity walkthroughs in archaeology sites. *6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*, 2005.
- [12] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London*, 1979.
- [13] Eileen Louise Vote. A new methodology for archaeological analysis - using visualization and interaction to explore spatial links in excavation data. *Thesis for the degree of Philosophiae Doctor (PhD) at Brown Computer Science - Rhode Island*, 2001.

Links

- [a] **Harris Matrix Composer**,
June 25th, 2014 - 3:40pm
<http://www.harrismatrixcomposer.com/>
- [b] **Unity3D Game Engine**,
July 31st, 2014 - 10:30am
<http://unity3d.com/>
- [c] **Ludwig Boltzmann Institut für Archäologische Prospektion und Virtuelle Archäologie**
June 25th, 2014 - 3:15pm
<http://archpro.lbg.ac.at/>
- [d] **GUI - File Browser by Daniel Brauer**
May 26th, 2014 - 2:20pm
<http://wiki.unity3d.com/index.php?title=ImprovedFileBrowser>
- [e] **Object Loader by Jon Martin**
May 30th, 2014 - 3:45pm
<http://www.jon-martin.com>
- [f] **GUI - Color Picker by Sergey Taraban**
May 23rd, 2014 - 10:35am
<http://staraban.com/en/simple-color-picker-control-for-unity/>