The Mayan *mascarón* from Chilonché (Petén, Guatemala):

New technologies for cultural heritage dissemination

Alessandro MERLO[1] | Eduardo VENDRELL-VIDAL[2] | Filippo FANTINI[3] | Carlos SÁNCHEZ-BELENGUER[2]

[1] DIDA: Dipartimento di Architettura, Università degli Studi di Firenze,Italy | [2] Instituto de Automática e Informática Industrial – ai2, Universidad Politécnica de Valencia, Spain | [3] Instituto de Restauración del Patrimoni, Universidad Politécnica de Valencia, Spain

Abstract: this paper describes the procedures used to achieve the first 3D reconstruction of a *mascarón*, discovered in 03.13.2009 in the archaeological site of Chilonché (Guatemalan Petén). This is a typical architectural decoration of Mayan buildings that represents a mythic animal (half jaguar, half reptile) with approximated dimensions of 4×3 meters and characterized by its good level of conservation.

The process started with data gathering (realized by an international team formed by Spanish, Guatemalan and Italian experts), carried out using a laser-scanner in an extremely narrow environment located in the substructure of the acropolis only accessible through the tunnel network (partially dug by looters). A high resolution polygonal B-Rep model has been obtained from the initial point cloud; furthermore high resolution pictures of the find have been projected on the model to add color information to the mesh. This extremely accurate and detailed model allows to study the topology of the original one and to replicate it using 3D printing technology. However, the excess of detail makes it unsuitable for real-time simulations. Using geometry simplification techniques, a low-poly model has been generated, integrating the geometrical information discarded from the original model through a normal map texture.

The narrow and uncomfortable network of tunnels of the site creates very difficult conditions for local population and tourists to visit the *mascarón*. Using the low-poly model generated, and real-time simulation software, two kinds of visualization applications have been developed: a first one intended to study the original object using virtual reality devices, and a second one intended to allow the general public to visualize the reconstruction over an interactive web-based 3D real-time simulation.

Keywords: real-time simulations, Mayan archaeology, laser scanning, mesh processing

General aspects and aims

The goal of this paper is to describe the pipeline developed in order to solve the problem of dissemination and promotion of a Mayan work of art hidden inside a pyramid forming part of the acropolis of the archaeological site of Chilonché (Guatemalan Petén, Fig.1).

Our first goal, as research team, was to provide 3D digital models with textures applied suitable for the visualization in common laptops, smart phones and tablets by means of a real-time application, at the same time we wanted to create a reliable asset for Unity 3D from the point of view of both geometrical-morphological accuracy and chromatic appearance (apparent color). The problem deals with a long series of different applications coming from distinct fields (laser scanner survey, mesh processing, photogrammetry and entertainment); each one of them presents power points and weaknesses, so it was important to take advantage from the strengths and avoid the usage of those parts of the applications that could not provide a

Fig. 1 – Chilonchè archaeological site and the Mayan *mascarón*

reliable solution from the point of view of the accuracy of the final result. In our opinion every modeling and texturing step should be intended as a part of an "experiment" where every phase aimed at geometrical decimation has to be controlled in terms of introduced error, avoiding an empiric approach to the creation of 3D digital contents, even if their purpose is to be visualized on the web.

The main scientific aspect of an experiment is its reproducibility that in this case has to deal with a heterogeneous set of data: high definition point clouds from phase shift device and a professional set of photos that had to be mapped on the surface of the 3D model. The methodology introduced in this paper explains how to use together digital and photographic survey in order to create the following outputs: real and scale models (by means of CNC equipment, or 3D printing solutions) from an high definition model, animations and still images through a medium resolution textured model, and a third output which is a low resolution textured model whose features allow it to be used inside game engines.

The survey campaign of the archaeological site of Chilonché

On the third of March 2009, an archaeological mission headed by the architect Gaspar Muñoz-Cosme of the Polytechnic University of Valencia and by the archaeologist Cristina Vidal-Lorenzo of the University of Valencia, that were operating at the archeological site of Chilonché in the region of Guatemalan Petén, found inside a network of tunnels a significant size statue representing a mythological creature and characterized by its good level of conservation. This statue, probably a lizard, presumably adorned the corner of the royal family palace, built in the original urban core of the Chilonché's acropolis.

Between 2010 and 2011 the research team documented with high accuracy the archaeological find through a traditional survey and a photographic campaign that produced 2D orthographic views.

ACQUISITION PARAMETERS OF THE *MASCARÓN*	
Equipment	**Faro Focus^{3D} S120**
Resolution	**1/5**
Quality	**4x**
Average distance between scanner and object	**60 cm**
Number of stations	**7**
Acquisition time:	**2h 30m**
Photo	**present**

Tab. 2 – laser scanner acquirement data

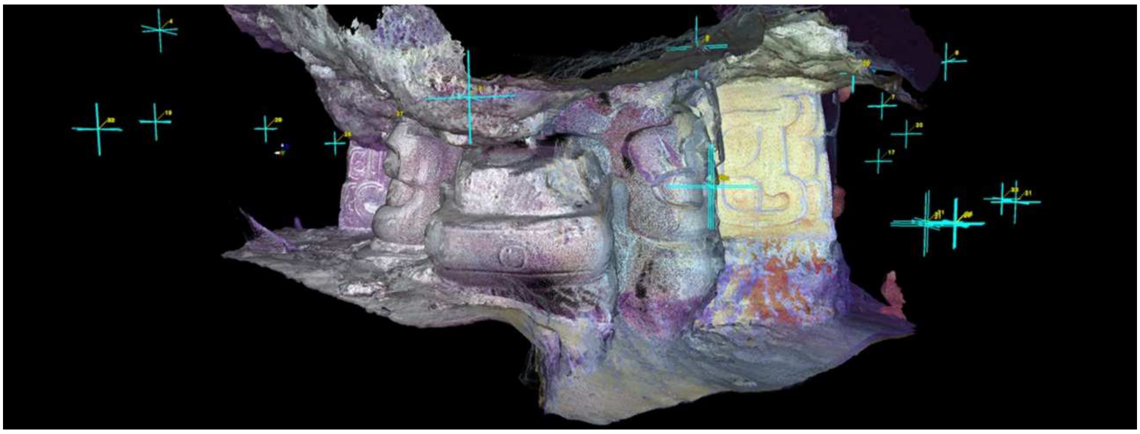

Fig. 3 – The point cloud model of *mascarón* without target

On March 2012, with the support of prof. Merlo (University of Florence) it was possible to make a first experimental survey using laser scanner technology in Chilonché (Tab. 2). The location of the sculpture, accessible only through a network of tunnels that surround it, and its complex morphology, lends itself well to the use of 3D scanning techniques. To avoid occlusions on the surface of the mask and, in order preserve its surface from deterioration, it was decided not to apply the targets on the zoomorphic Mayan sculpture, but rather stick them on the walls of the cave (Fig. 3).

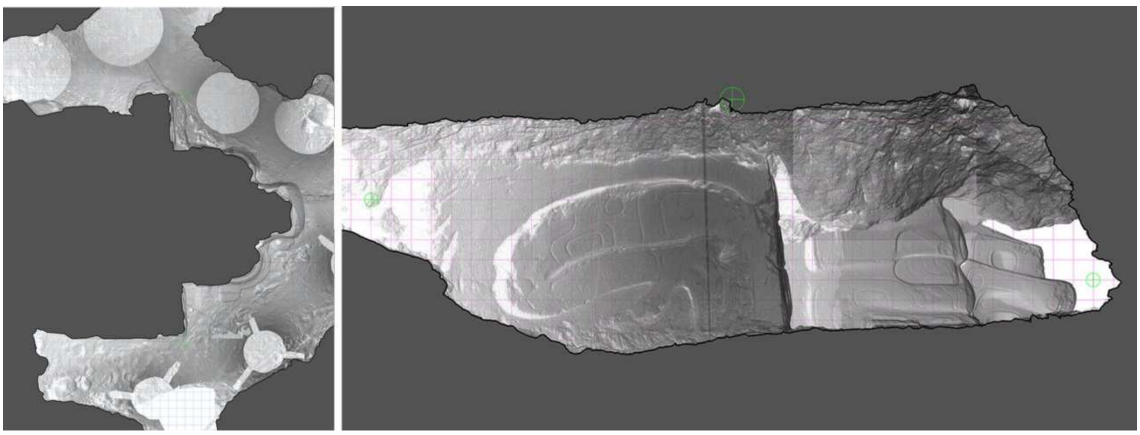

Fig. 4 – The point cloud model of *mascarón* without target

The first phase of the study was to register the 7 scans carried out. From the cloud model were obtained plans and elevations that allowed us to check previous surveys (Fig. 4).

Modeling and texturing of the Mayan *mascarón* from Chilonché

The adopted equipment is a long range scanner in use for surveying architectures and archaeological sites, but in this case it produced good results even if it wasn't a mid or close range apparatus such as a triangulation or laser stripe equipment (the geometric input data from laser scanner survey of the *mascarón* are summarized in the table 5).

POINT CLOUD MODEL	
Total number of points	199×10^6
Global accuracy of the registered model	2 mm
Size of the .imp database:	3368 Mbyte

Tab. 5 – Laser scanner data: quantitative aspects

The density of the sampling is quite high for the description of an object of such measures (approximately 3 x 4 meters); the average distance between points calculated on the whole set of point clouds is 1.936 mm.

The set of scans is composed by seven point clouds whose quantitative aspects are summarized in the illustrations shown in Fig. 6; the optical features of the material (sedimentary stone) can be considered optimal for data acquisition with a phase shift device at the contrary in hypothetic case of a *mascarón* made of marble the scattering effect due to an high translucency level would have badly affected the quality of survey producing an higher level of noise and relevant speckle effect (AMORUSO, APOLLONIO, REMONDINO, 2010: 119-161).

The meshing phase was carried out on a fenced version of the point cloud because the aim was to represent the Mayan sculpture without the occlusion produced by tunnels. The result of the merging/integration of all the scans is shown in Fig. 7 where, even the at first sight, can be perceived a relevant level of noise of the mesh model but, at the same time, a good coverage of the *mascarón* surface without any big hole.

The standard procedure implemented inside the majority of mesh processing application is aimed at "healing" the mesh from all topological errors as dangling faces, small polygons clusters, non-manifold faces, etc.: also, in this case, we adopted this procedures on the heavy mesh made of 39,558,677 polygons, but the main problem to solve was the absence of thickness of the whole model. It was necessary to build and digitally "construct" the pedestal on which the surveyed *mascarón* could lay and, obviously, reverse modeling applications do not provide enough flexibility to manage such problems. Also the irregularity of the profile of the upper part of the *mascarón* needed an accurate geometric modeling approach inside NURBS or mesh modeling applications (Fig. 8).

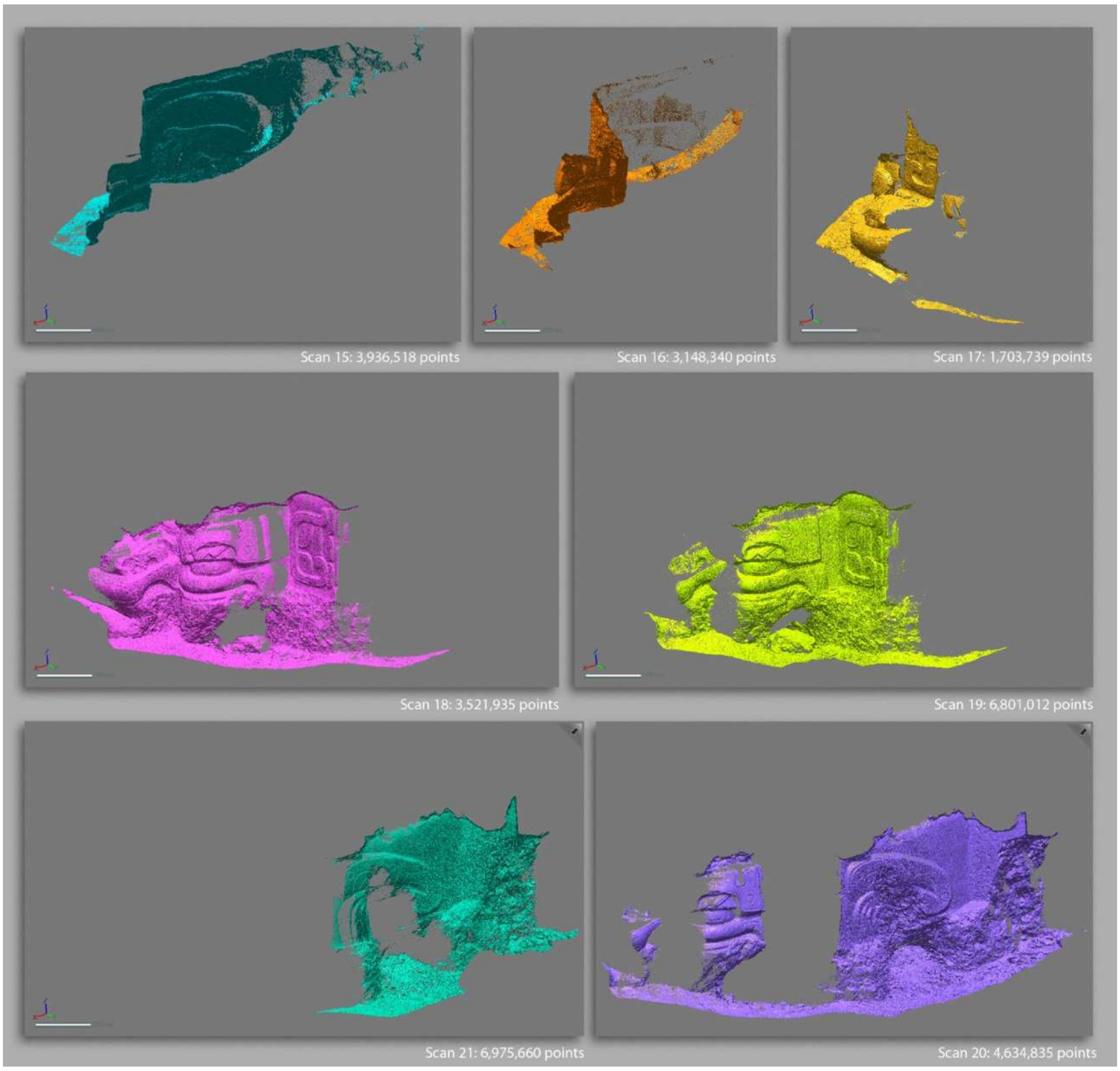

Fig. 6 – The set of scans acquired by means of phase shift laser scanner equipment

Thanks to Rapidform XOR powerful global remeshing tools it was possible to resample the whole mesh before any further modeling phase; these kind of commands improve the mesh quality by calculating a new mesh whose triangle edges are nearly equal to an average measure calculated by the program, or directly expressed by the user. The resulting isotropic mesh is smoother than the original one and characterized by a better connectivity of triangles and, even if it cannot be considered a structured one, it produces good shading results once rendered. In Tab. 9 are listed, for each model, the associated number of polygons and space occupied by the files on the disk.

For the integration of missing parts of the model we opted for a geometric modeling application and, in particular, one that supported subdivision surfaces because, thanks to their flexibility and variable level of detail, can be easily modeled using a low-poly control cage (FANTINI, 2012) and then integrated with the globally remeshed model from active sensor.

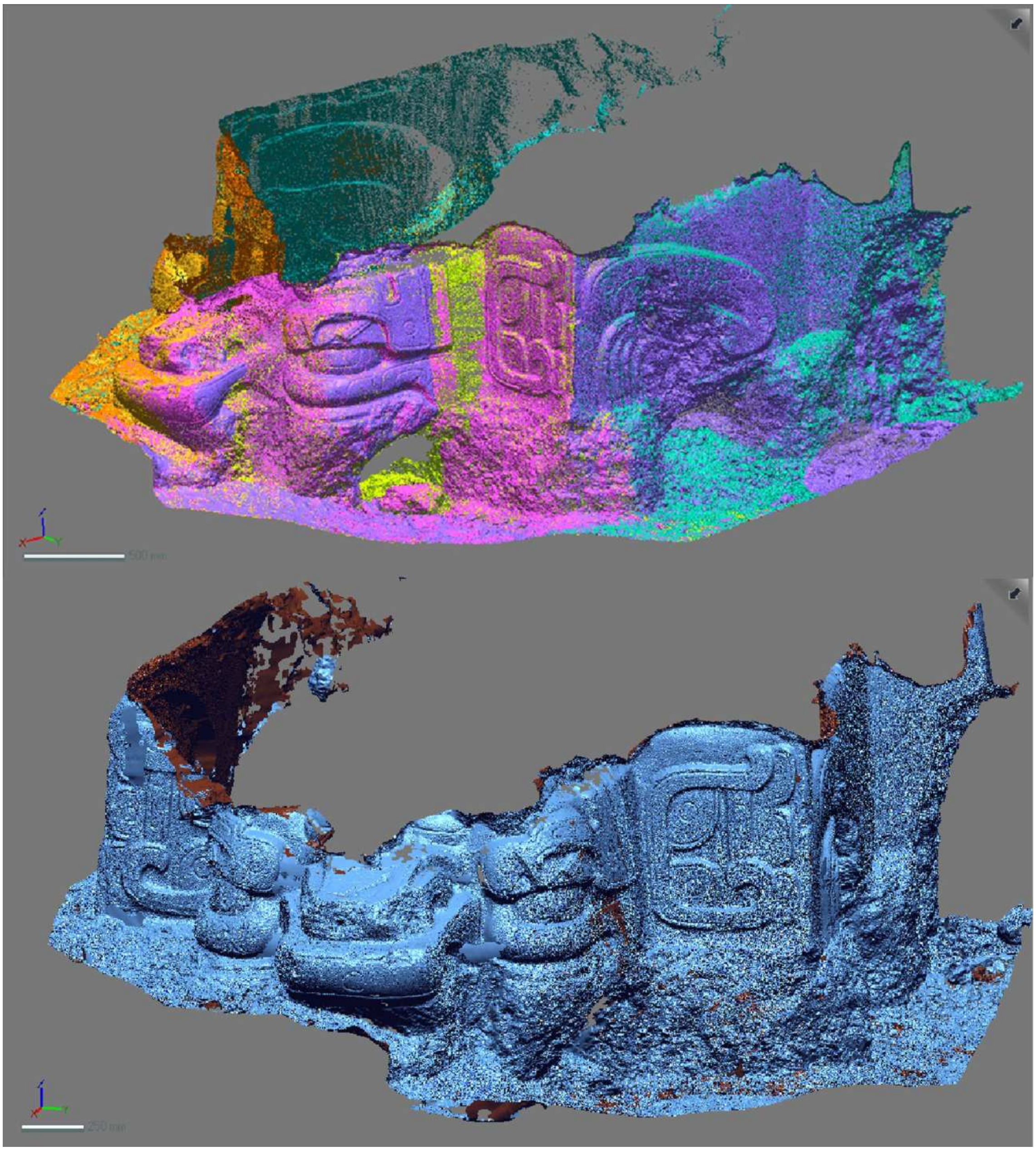

Fig. 7 – The point cloud set and the result of the meshing phase

Catmull-Clark subdivision surfaces can be easily converted into models with the user's required number of polygons and, in this case, the bi-cubic patches have been frozen into a dense structured mesh with edges of the same average measure of the globally remeshed model. Using conventional bridge and "fill-holes" tools inside Rapidform we obtained a 2-manifold B-Rep (Fig. 11).

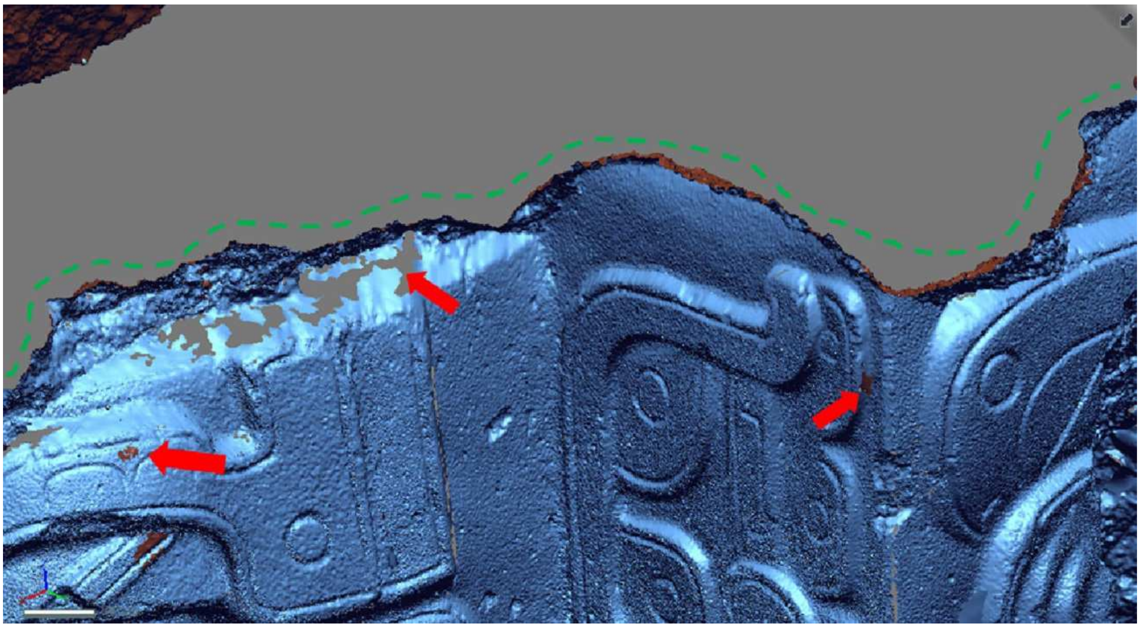

Fig. 8 – Some smaller topological problems can be easily managed inside the reverse modeling application but, the irregular hole on the top and bottom of the *mascarón* needs a direct modeling approach

POLYGONAL MODEL				
	High poly	**Global remesh**	**Mid-poly**	**Low poly**
Number of polygons:	$39x10^6$	$5x10^6$	163,494	40502
Size of the .xrl file:	789.287 Mbyte	210.614 Mbyte	13,779 Kbyte	953 Kbyte

Tab. 9 – Mesh model reduction and quantitative aspects

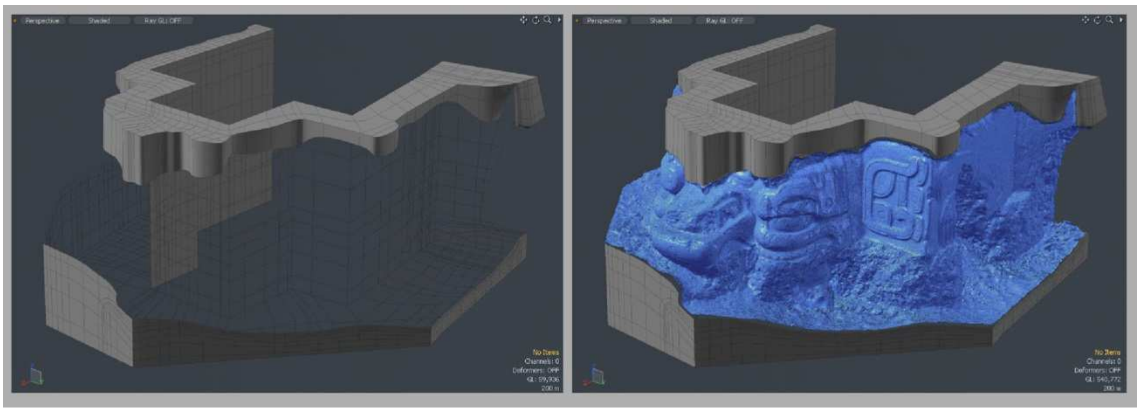

Fig. 10 – External cage created through subdivision surface modelling. It is important to notice the capability of Catmull-Clark extended to create sharp creases without having to increase the number of polygons

The topological accuracy of this model allow to use it inside CNC tools for the production of a real scale replica of the *mascarón* which can be considered the most proper solution for the visitors of Chilonché given that the underground maze of tunnels quite dangerous for not trained tourists. In any case, this high-poly model does not match with a “confortable” UV mapping inside entertainment applications because it is still too heavy (an unwrapping is not advisable). Consequently, being real-time visualization and animations

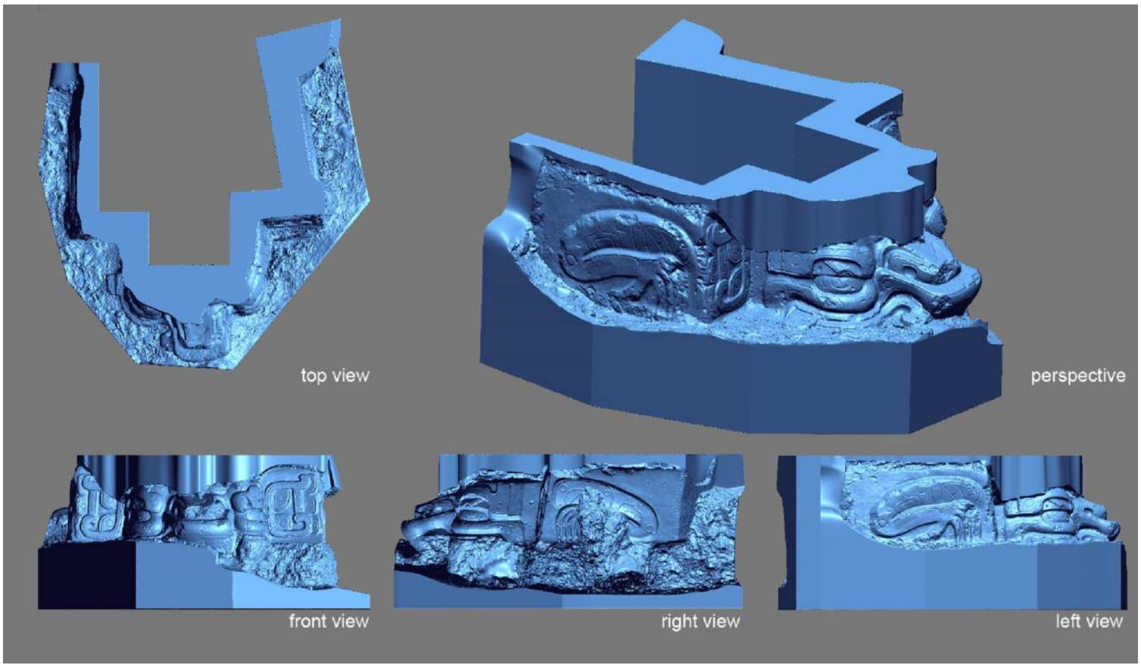

Fig. 11 – Orthographic views of the final high-poly model

within the goals of our project, we applied a standard procedure based on relevant reduction of polygons by means of a decimation tools capable to take into consideration the model curvature (optimize mesh command), and then we unwrapped the model inside Luxology Modo. This program also provides useful tools aimed at calculating normal maps by means of render-to texture solution: the normals of the high-poly model were encoded inside the UV reference system of the optimized mesh (mid-poly: 163,494 polygons). The next step is to provide to this model a correct apparent color texture, applied by means of the same UV system used for the normal map that, in this case, also provides a good reference for understanding how reliable was the mapping of images on the model's surface.

The projection of an un-oriented set of eleventh images on the mid-poly model was achieved thanks to EOS System Photomodeler, a popular photogrammetry application, also capable to calculate the camera resectioning for each photo, knowing the position in the space of a sufficient number of homologous points (between the picture and the object). We picked a number of approximately 10 vertices (natural targets) for each picture inside Rapidform (add-reference point) and, at the same time, we marked the same points on the photos (Fig. 12).

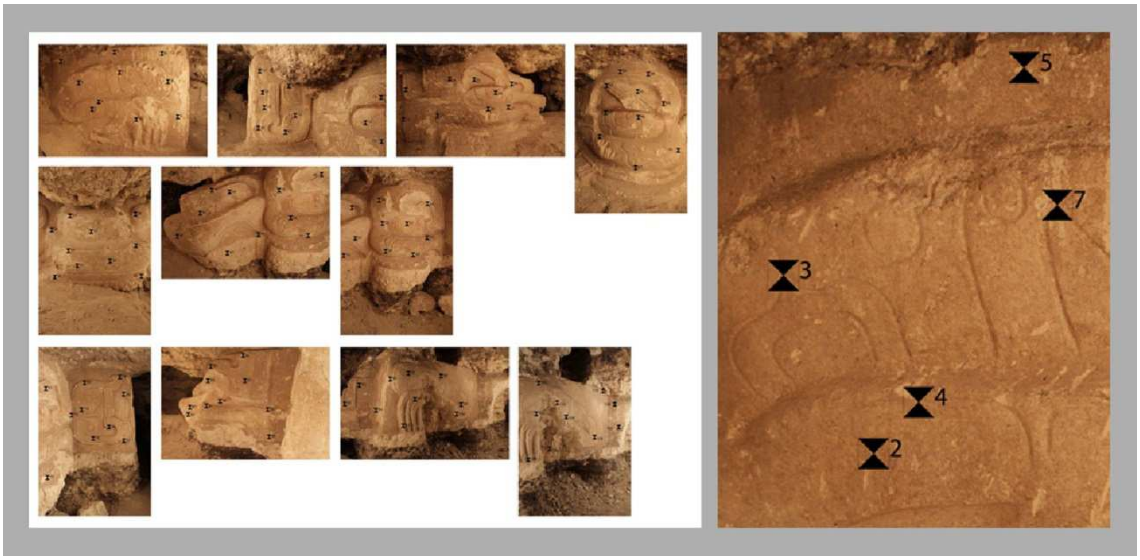

Fig. 12 – The un-oriented set of photos with targets marked

Once exported from Rapidform using the DXF file format, point sets can be introduced as reference inside Photomodeler: by picking 3D vertices and the corresponding pixels on the photos, the program orients a camera for each photo. Once obtained the position of every camera it is possible to export them inside Luxology Modo using the FBX file format and then re-project the images on the mid-poly model (Fig. 13).

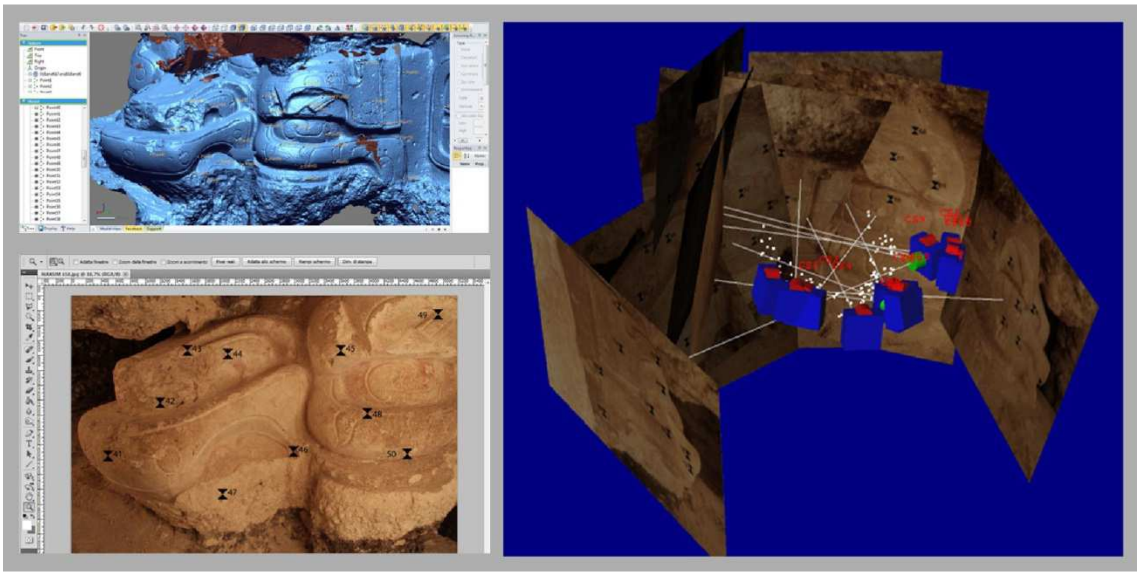

Fig. 13 – Homologous points marking in 3D and 2D and the result of camera resection inside Photomodeler

The resulting model can be used inside different rendering applications being normal and color maps (Fig. 14) applied by means a standard UV reference system.

Fig. 14 – Homologous

On-line Realtime Visualization

The main goal of producing an on-line 3D simulation is to disseminate tangible cultural heritage through the Internet. By using this technology, we are able to show users all over the world the generated 3D model of the *mascarón*.

To implement the proposed simulation, we chose Unity3D as the rendering engine. The main reason is because it allows generating simulations for almost every device with enough graphics performance (PC, Linux and Mac computers, Android and iOS devices...) and, mainly, because it allows displaying the simulations embedded in a web page.

Originally, Unity was conceived as a 3D game engine but, since it has evolved so fast in the last years, nowadays it can be used for almost every kind of 3D project (game or not). It provides great graphics performance, supporting modern GPU’s (Graphical Processor Unit) capabilities.

Hardware Architecture

Since on-line visualization happens in a networked environment, the architecture of the application is based on a client-server paradigm with a Software Delivery strategy. There are four main components that play specific roles, and which are illustrated in Figure 15:

- Application server: stores 3D data, information and the application code, and transfers them to the clients. Since it serves to several clients, it does not provide complex computations for individuals. Instead of this, the server sends the application code and the data to the client, in order to generate the real-time simulation.
- Player server: stores the binary files that perform the web player installation on client computers. Since this web-application requires direct access to graphic hardware in order to perform a fast and complex simulation, clients have to install a player in their browsers that loads the code stored in the application server, and performs the simulation.

- Client: in charge of the real-time rendering, interaction with the user, and data requests to the server. First application run requires a player installation through the player server.
- Network: responsible for data transfers between client and server. Because of time delays, interactions between the user and the 3D environment have to be performed primarily on the client side, keeping this client-server communication only for data requests.

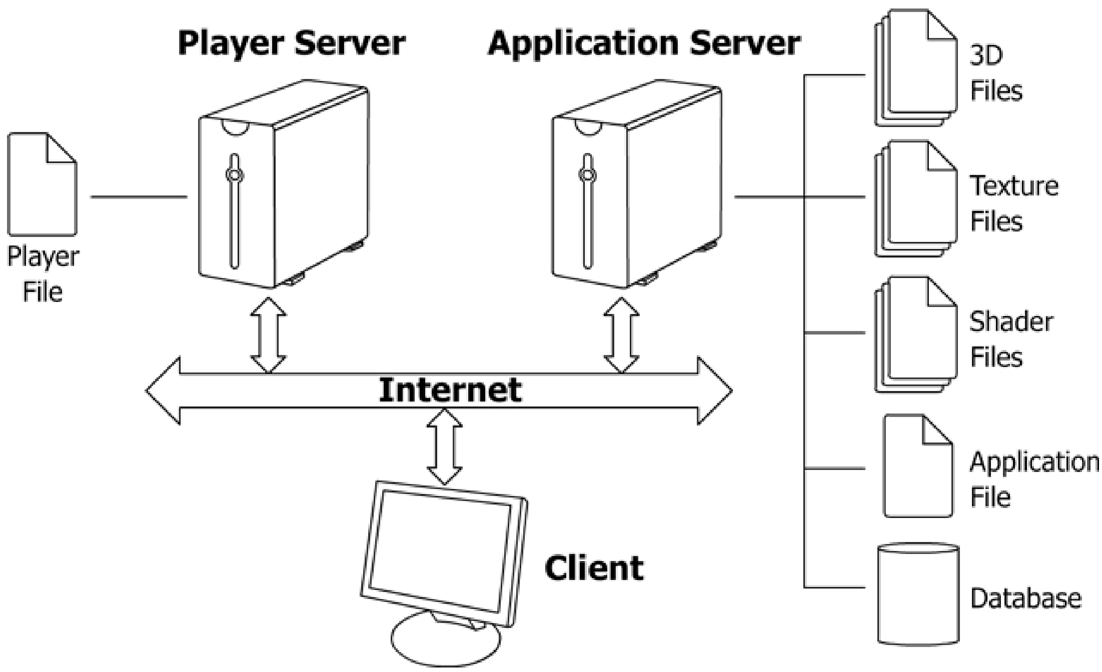

Fig. 15 – Client/Server hardware architecture of the proposed on-line 3D simulation

Challenges

The main challenge when creating real-time 3D visualizations is the duality between realism and performance: a realistic visualization requires high resolution textures and high density models to be accurate, whilst a visualization focused in performance needs to over-simplify the represented data in order to accelerate calculations.

When creating a simulation on a networked environment, a new restriction is added: the required bandwidth for downloading 3D models and textures can be so high, that a major bottleneck appears in data transfers between the client and the server.

It also has to be taken into account that, when publishing an on-line 3D simulation, the potential consumers of the developed contents present a heterogeneous set of computers performance.

The computational cost of rendering in real-time depends on three major aspects:

a) The resolution of the displayed textures.
b) The polygonal density of the 3D models.
c) The visualization enhancements achieved by using complex shaders for polygon-shading, shadow casting, post-processing effects…

The two first aspects are related to the displayed data, and also affect the network bandwidth usage, whilst the third one is network-independent.
Considering all these restrictions, we have focused the development of the proposed visualization according to two premises: (1) it has to provide correct results for low performance computers, and (2) it has to provide realistic results for high performance computers.
The direct consequence of the stated premises is the need to develop several LODs (Levels of Detail) for the displayed data. Then, the proper quality level that best fits the rendering performance of the machine that is running the simulation has to be selected, either by the user or automatically by the application.

Texture Levels of Detail

Memory and bandwidth are the major problems related to textures: as the texture resolution increases, the amount of data to store and transfer increases exponentially. This fact can be critical when displaying a large amount of textures in a low memory GPU, or a slow Internet connection computer.
Luckily, dealing with this is not complicated: since textures are flat matrices of bits representing contiguous color values, automatically scaling them is trivial. This way, generating multiple texture Levels of Detail is not a big problem.
The resulting representations of the same texture are stored separately into the server so, once a client has chosen the desired quality level to display, only the appropriate files will be transferred, ensuring that the minimum necessary bandwidth and memory are spent.
Figure 16 shows the results achieved in the developed simulation using different texture qualities for the same model.

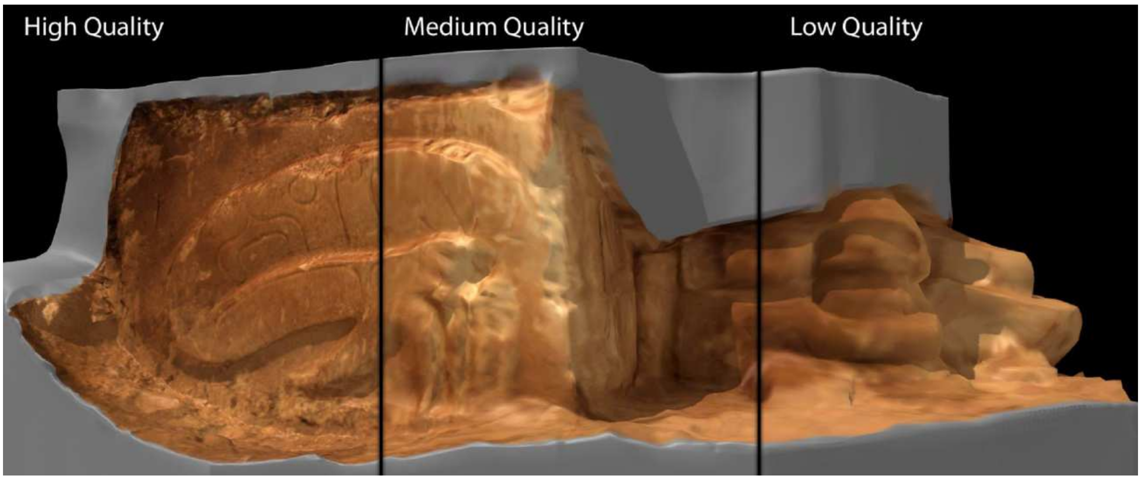

Fig. 16 – Multiple Levels of Detail for textures in the developed simulation

Geometry Levels of Detail

As happened with textures, the polygonal density of 3D models affects both bandwidth and memory requirements. In addition, the computational cost of displaying a dense mesh also represents an important extra load for the computer.

The main difference between geometric LODs and texture LODs is that, in this case, the simplification process is very hard to automate: only state-of-the-art GPUs support efficient dynamic geometry subdivision, so it cannot be assumed that the final user will be able to use these techniques. This means that the same model has to be manually decimated several times, producing as many meshes as LODs we want to use. Once geometric LODs have been created, there are two different ways of using them: (1) once the user or the application has selected the proper quality level for the simulation, only the corresponding mesh is downloaded and displayed, giving a constant level of detail visualization, or (2) as the viewpoint approaches a model, more refined LODs are downloaded and displayed and, as the viewpoint gets far from the model, more simple LODs are displayed.
The first approach ensures minimum hardware requirements, whilst the second one provides a more realistic visualization, saving resources when the model is so far that no details can be appreciated. Figure 17 illustrates two different geometric LODs for the same model.

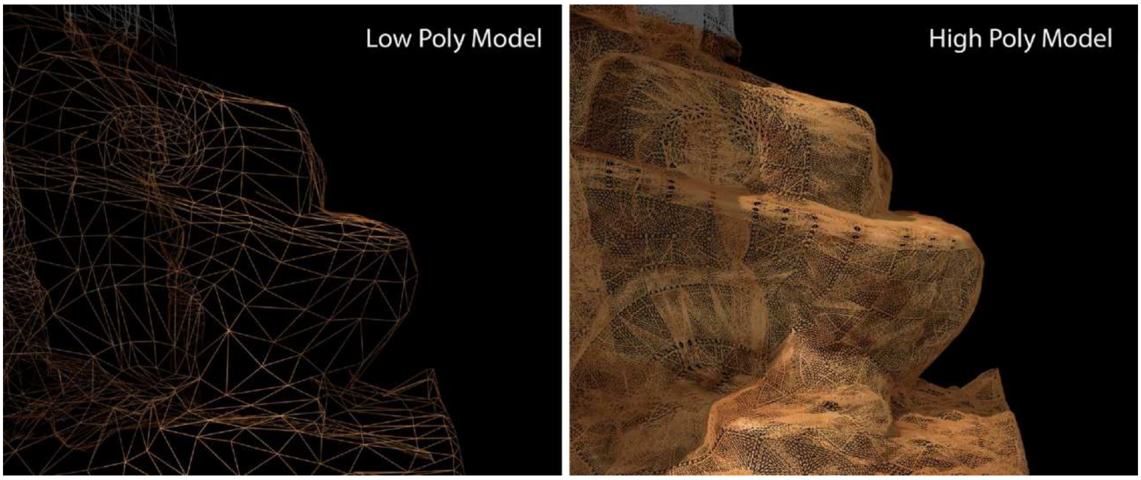

Fig. 17 – Two different Levels of Detail for geometry in the developed simulation

An alternative way to keep low the amount of polygons in the displayed model, while providing realistic shading, consists in using a low-poly mesh with a normal map texture to increase detail.
Normal mapping technique is very common in real-time 3D simulations, and it is used for faking the lighting on the surface of the model. With normal mapping, details can be added without using more polygons. A common use of this technique is to greatly enhance the appearance and details of a low polygon model. Normal maps are commonly stored as regular RGB images where the RGB components corresponds to the X, Y, and Z coordinates, respectively, of the surface normal.
Since all normal mapping calculations happen in the *Pixel Shader Stage*, in the graphics pipeline, this technique does not have the capability to refine the model: the amount of vertices of the resulting model is always constant.
As shown in Figure 18, results using this technique are much better than results achieved with a simple lightning, with the only cost of one extra texture.

Fig. 18 – Comparison between diffuse lightning (left) and normal mapping results (right) in the developed simulation

CAVE Realtime Visualization

Once 3D models and real-time simulation have been created, the resulting application has been ported to a CAVE (Cave Automatic Virtual Environment), which is an immersive virtual reality environment where users can interact with the model by using a natural interface: to navigate around the *mascarón* it is only necessary to physically move inside the simulation space.

This interaction interface is very interesting for researching purposes because the end-user does not need any kind of ground knowledge in complicated 3D interfaces to fully visualize the model. It is also interesting because, since the simulation is immersive, the user gets fully surrounded by the environment and, since it is an augmented reality simulation, the user can also see himself. This way, the scale sensation when navigating inside a virtual environment makes the simulation much more descriptive than a regular visualization displayed in a traditional screen.

The CAVE simulation environment we have used is made up of four rear-projected displays (one in the floor and three vertical in the sides), a shutter 3D glasses that synchronize with the refresh of the displays to create an immersive 3D stereoscopic visualization, a tracking device made up with 4 infrared cameras located in the ceiling of the room that allows calculating the position of the user inside the simulation space and a powerful computer (AMD Opteron with 8 GB of RAM) connected to a nVidia Quadro Plex adaptor that allows us to simultaneously use 4 nVidia Quadro FX4500 GPUs.

Given the computational performance offered by the hardware, high definition textures and high poly models of the *mascarón* have been used in order to create a very realistic visualization.

However, working in a CAVE environment presents an extra rendering cost: since we work in a stereoscopic 4-display environment, for each simulation frame 8 images have to be calculated: one for each eye, for each display. This means that the rendering time is eight times greater than the rendering time of a traditional simulation.

The typical execution cycle for each rendered frame in a CAVE simulation is the next:

1. Read the input from the tracking cameras, and calculate the position of each eye of the user in the physical simulation space.
2. Calculate the image to display, for each screen, according to the position of the left eye, and write it into the quad buffer of the graphics card.
3. Calculate the image to display, for each screen, according to the position of the right eye, and write it into the quad buffer of the graphics card.

Then, the graphics hardware automatically synchronizes the four displays to show only the left eye images, while occluding the user the right eye in the shutter glasses and, afterwards, shows the right eye images, while occluding the user the left eye. This way, the stereoscopic sensation is achieved.
To simulate the immersive experience, the image shown in each display must be calculated by correcting the projection matrix of the 3D camera according to the user's eye relative position respect each display. These calculations are, however, very fast to perform, being the main performance bottleneck the need to generate eight images every simulation frame.
Figure 19 shows the resulting simulation inside the cave, and the rear-projected displays used.

Fig. 19 – (left) Results of the developed simulation inside the CAVE. (right) Rear-projected displays used, with a mirror to reduce the room space requirements

Acknowledgements

This project has been partially financed by the "Programa de Ayudas de Investigación y Desarrollo (PAID)" of the Universitat Politècnica de València and the "Plan Nacional de I+D+i 2008-2011" from the Ministerio de Economía y Competitividad of Spain, Project ID: HAR2012-38391-C02-02.

References

AMORUSO, G., APOLLONIO, F.I., REMONDINO, F. (2010) Caratterizzazione strumentale di sensori attivi a tempo di volo e a triangolazione. Utilizzo di laser scanner su superfici marmoree di epoca romana. In Benedetti B., Gaiani M., Remondino F. (ed.) Modelli digitali 3D in archeologia: il caso di Pompei. Pisa: Edizioni della Normale.

BRODLIE, K. (2000) Visualization over the World Wide Web. Scientific Visualization (Dagstuhl '97 proc.), IEEE Computer Society Press, pp.23-29.

CHITTARO, L., RANON, R. (2007) Adaptive 3D Web Sites. Brusilovsky, P., Kobsa, A., Nejdl, W. (editors), The Adaptive Web - Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321, pp. 433-464.

CRUZ-NEIRA, C., SANDIN D. J., DEFANTI T. A. (1993) Surround- screen projection-based virtual reality: The design and implementation of the cave. Proceedings of SIGGRAPH 93, Computer Graphics Proceedings, Annual Conference Series, pages 135–142.

FANTINI, F. (2012) Modelos con nivel de detalle variable, aplicados a la arqueologìa, realizados mediante levantamiento digital

In EGA, N° 19, pp. 306-317.

LEE H., TATEYAMA Y., OGI T. (2010) Realistic visual environment for immersive projection display system. Virtual Systems and Multimedia (VSMM), 2010 16th International Conference on, pp. 128 – 132.

MERLO A., DALCO' L., FANTINI F. (2012) Game engine for cultural heritage. new opportunities in the relation between simplified models and database. Virtual Systems and Multimedia (VSMM), 2012 18th International Conference on, pp. 623-628.

PAPAIOANNOU, G, et al (2003) Enhancing Virtual Reality Walkthroughs of Archaeological Sites. Proceedings of the Eurographics 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, pp. 193-201.

YAPO, T., SHENG Y., NASMAN, J., DOLCE, A., LI, E., CUTLER, B. (2010) Dynamic projection environments for immersive visualization. Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 1–8.

Unity – Game Engine. http://www.unity3d.com. San Francisco, United States. (13 Jan. 2013)

ZARA, J. (2004) Virtual Reality and Cultural Heritage on the Web. Proceedings of the 7th International Conference on computer Graphics and Artificial Inteligence. Limognes, France, pp. 101-112.

ZARA, J. (2006) Web-Based Historical City Walks: Advances and Bottlenecks. PRESENCE: Teleoperators and Virtual Environments. Vol. 15, No. 3, pp. 262-277.

Imprint:

Proceedings of the 17th International Conference on Cultural Heritage and New Technologies 2012 (CHNT 17, 2012)

Vienna 2013

http://www.chnt.at/proceedings-chnt-17/

ISBN 978-3-200-03281-1

Editor/Publisher**:** Museen der Stadt Wien – Stadtarchäologie

Editorial Team: Wolfgang Börner, Susanne Uhlirz