# Remote Rendering and Visualization of Large Textured 3D Models

Dante Abate, Silvio Migliori, Samuele Pierattini
Unit on Informatics System Development and ICT
ENEA Research Centre
Bologna, Italy
{dante.abate, silvio.migliori, samuele.pierattini}@enea.it

Belén Jiménez Fenández-Palacios, Alessandro Rizzi, Fabio Remondino
3D Optical Metrology Unit
Bruno Kessler Foundation (FBK)
Trento, Italy
{bjfernandez, rizziale, remondino}@fbk.eu

*Abstract* — **There is an increasingly urgent need, expressed by the scientific community and by the end users, to exploit web resources in order to analyse, process, visualize and interact remotely and in real time with 3D high definition models of Cultural Heritage artefacts and monuments. Many sharing platforms are currently developing thanks to user contributions. However this goal has not been yet reached successfully. Complex mesh models cannot be displayed interactively online on consumer computers, without downloading locally the entire model or accepting the compromise solution that the model is blurred and fuzzy during the interaction. Thus a key aspect is the quality with which the content is "distributed", and, at the same time, the certainty of copyright protection. The platform implemented and here presented allows an efficient and effective multi-user online sharing of high quality 3D textured models without the need for the users to download it locally but exploiting the performances of a remote HPC infrastructure.**

**Keywords -** ***Remote Rendering, High Definition Mesh Models***

## I. INTRODUCTION

Thanks to the latest technological developments at hardware and software level, nowadays it is quite promising the possibility to visualize remotely and online large and complex 3D digital models, possibly linked to external databases for analyses and educational purposes. So far the presence of 3D models on the web is not very frequent, despite the availability of some standards and the increasing developments of libraries (C3DL, OpenSceneGraph, X3DOM, etc.), API (WebGL, O3D etc.), plug-in applications (OSG4Web, Java3D, etc.), game engines (Unity3D, 3DVia, etc.) and languages and technologies (VRML, X3D, PDF3D, etc.) [1]. Nevertheless the scientific community and the end users are expressing an increasing need to exploit 3D models with a client-server architecture. But when it comes to the visualization and interaction of big polygonal datasets online, the reliable commercial solutions are weak. Probably the most well-know and performing web-based rendering approach for 3D models (terrain and man-made structures) is provided by GoogleEarth although it does not allow to display and interact large 3D polygonal models. The common solutions often stall the image/model display and generation until the entire geometry's download in the cash is completed. The main problem is probably the bandwidth of the network although even the protection of online shared and rendered 3D models is still an open issue. Indeed sharing online digital archives of 3D (Cultural Heritage) models presents new challenges for the protection of intellectual property rights (IPR). Different approaches [2, 3, 4, 5] were proposed to protect piracy, copies and misuse - still allowing an interactive sharing – but the topic is still under investigation.

The article presents the study conducted at the ENEA research centre - unit on Informatics System Development and ICT - regarding the implementation of a hardware-software architecture which permits a multi-user access to a repository of 3D models exploiting different applications according with different goals [6]. With respect to other approaches, the end user no longer needs specific hardware and software resources during the interaction and visualization of a 3D model whose resolution remains constant during the navigation. At the same time the system protects the intellectual property related to the 3D model since they are not downloaded locally but they remain safely stored in the remote repository. All the potentialities of the remote HPC infrastructure (CPU and GPU) are exploited using a simple Java applet which can be installed on proposed threads.

## II. RELATED WORKS

In the past years the scientific community focused on the rendering and visualization, in real-time and possibly via web, of large 3D models of the environment (e.g. terrain models) or urban areas [7, 8, 9, 10]. Generally a hierarchical and local representation of geometry and texture is employed (LOD approach). A simpler geometric representations for far geometries is coupled with impostor-like techniques, ray-tracing and GPU programming for the textures representation [11, 12, 13]. Thus many remote rendering platforms have been developed with different rendering approaches. Besides simple applications which allow the visualization of low resolution 3D models on the web, the most interesting applications are those aiming at visualize and interact with large mesh models.

An image-based LOD method based on a server-client approach was proposed by Koller [2]. In this approach, the server has the full resolution geometric model, the user (client) has a low resolution model. The server simply renders locally and then streams images corresponding to the request from the user who is interacting with the reduced model. This is the approach developed in ScanView [14], a client / server rendering system for viewing complex 3D models. The client side of ScanView consists of a freely available viewer program and a set of simplified 3D models. Using the viewer, a user can interact with a simplified model. The server sends high-resolution images to the client only when the users stops to interact. A typical query-response cycle takes some seconds depending on network traffic, the speed of the connection and the current load on the server. The ScanView system incorporates a number of defences with respect to IPR and piracy. The Digital Sculpture Project [15] is also based on the ScanView system while a very similar architecture is in the Venus 3D model publishing system [16]. It is an interactive publishing system for large 3D models (millions of polygons) on the web exploiting an image based approach. Currently it's available for web standard browsers which support WebGL and for Mac mobile devices (Apple iPhone/iTouch and iPad).

A point-based representation was presented in [17, 18] where the rendering system use points as the rendering primitive with a hierarchal and LOD data structure. Point-sampled objects do neither have to store nor to maintain globally consistent topological information. Therefore they are more flexible compared to triangle meshes when it comes to handling highly complex shapes. Depending on the complexity of the input model, the communication traffic between server and client can still be high. This approach was implemented in the QSplat visualization program which can be used locally or in a network-streaming version [19]. An efficient framework for remote or mobile devices by means of QSplats is described in [20].

The mesh-based method represents 3D objects with mesh models in multi-resolution. For example, the Adaptive Tetra-puzzles proposed in [21] converts the input mesh into a hierarchy structure composed of nodes, containing smaller meshes. The method uses a regular conformal hierarchy of tetrahedra to spatially partition the model. However sending mesh data over the network can still be very expensive.

A mixed approach of image and mesh-based rendering requires the server, where the geometrical model is stored, to pre-render the mesh models from various viewing positions, and stores, at the same time, these images in a repository [22, 23]. The client requests to display the model from specified viewpoints. The server send back the pre-rendered images necessary to calculate the view as well the sparse mesh model.

The SAVE (Serving and Archiving Virtual Environments) project [24] aims to create an on-line, peer-reviewed journal where authors can publish 3D digital models of Cultural Heritage sites and monuments. Its goal is to offer scholars in the field of virtual archaeology and architectural history the opportunity to publish on the internet 3D digital reconstruction of artefacts, monuments and settlements as fully interactive 3D models. These data are associated to some documentation giving transparency to the end-user about the hypothesis of reconstruction of the major elements of the digital model.

Some thoughts on centralized 3D archives, 3D Cultural Heritage repositories, IPR and remote visualization are given in [25].

## III. INFRASTRUCTURE

The project, called ARK3D [26], uses the ICT infrastructure of ENEA-GRID and in particular the graphic cluster which belongs to the project CRESCO housed at the ENEA Research Centre of Portici (Napoli, Italy). CRESCO High-Performance Computing (HPC) infrastructure, included in the top500 supercomputers list [27], is equipped with a cluster divided into sections based on hardware features (CPU, RAM, Graphic Cards) and it has a total peak of 28 TFLOPS. To date, the graphics section consists of 12 workstation with AMD dual core processors, 16 GB RAM (nodes), each one equipped with NVidia Quadro FX graphic cards, for a total of 68 cores. The storage system, with a centralized system for data backup, provides direct access both to the geographical file system AFS and GPFS (General Parallel File System – High Speed Storage 2 GByte/s and 160 TByte). The GARR network (Italian Academic & Research Network) provides the internet connection with two lines of 1000 Mbits.

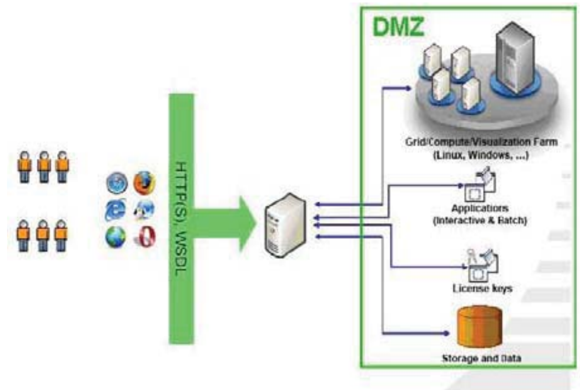



Figure 1: ARK3D Architecture

## IV. ARCHITECTURE

### A. Repository

The AFS geographic file system is used to implement the 3D repository (Fig. 1). 3D models can be uploaded together with different kind of metadata by users through a web based interface following a dedicated registration procedure. The database is queried via web by free search keywords. The result will contain the metadata attached to the models, together with a link to the 3D remote application. The rendering (Fig. 2) is performed using the dedicated remote cluster and it guarantees the protection of the data which can be manipulated but not downloaded, if not through a secure ftp. The whole architecture is scalable, both in the number of models uploaded and in the number of simultaneous users logged in.

### B. Repository query

The database is queried via a web page. The 3D models are divided into specific projects created by the authors. Unregistered users can only visualize an image (screenshot) of a 3D model. On the other hand, users who have obtained credentials from the system administrator, have the option to choose and run the remote application which allows to visualize and analyse the selected 3D model.


Figure 2: Remote 3D rendering based on Cresco graphic section.

### C. Client

The user, through a dedicated web page, has to run a java applet which installs and automatically configures the client to access the selected graphic application (Fig. 3). The user is not required to know any configuration feature or install other kind of software. Between the client and the server the data transfer is exclusively represented by a stream of compressed images, generated remotely by the application on the HPC infrastructure, together with the functions of keyboard/mouse interaction. Operations delegated to the client program are limited to images decompression and keyboard/mouse input management. With this kind of technology the final user can work over very large 3D models and visualize them (point cloud mode, shaded or textured) using standard consuming computers, just connecting via web to the remote cluster.


Figure 3: Client Java Applet.

### D. Security

AFS file system stores an Access Control List (ACL) for each directory. It defines which users and groups can access the directory, the contained files and which operation they can execute. Usually, when a user logs in into the system through a Kerberos 5 authentication system (standard network protocol for providing secure communications over insecure networks), he/she obtains an AFS token in order to work on the GRID and run only operations which are specified in the ACLs. ARK3D architecture provides a further level of security. All ACLs have been reset and, at the same time, a Keytab (certificate) has been created for each specific application which is supposed to run the 3D models. This basically means that only that specific application can access the folder containing those 3D models.

### E. Upload system

The user, through the dedicated web page can upload 3D data onto the repository, filling in different fields, some of them mandatory, to complete the task. Besides the geometry (point clouds or polygons) and textures, the user can eventually upload metadata attached to the 3D object. The metadata are represented by basic information like model's author, upload/creation date, file name, etc., either in form of strings of characters or independent files like images or documents (pdf, doc, jpeg, etc.).

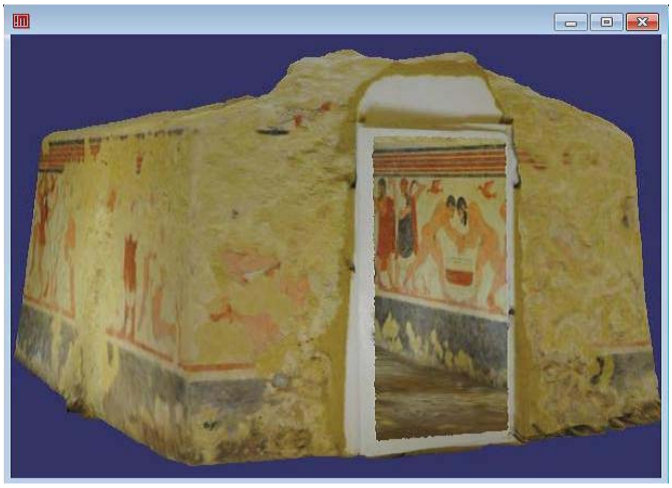


Figure 4: Example of remote rendering of a large polygonal textured 3D model, visualized on the client using OSG Viewer. The texture size is ca 2 GB.

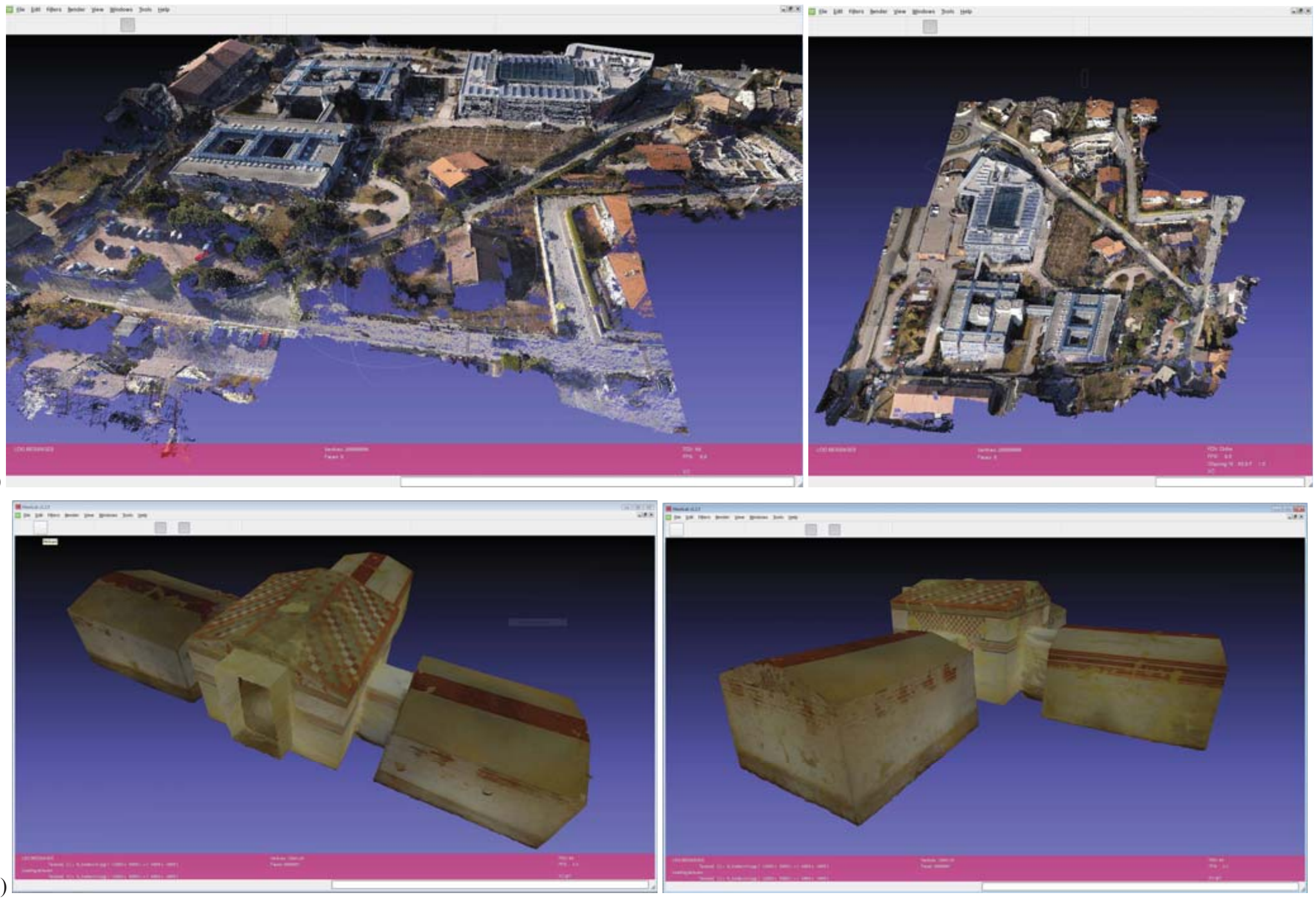
## V. VISUALIZATION

### A. Simple viewer

Currently the standard viewer is implemented using the open source high performance OpenSceneGraph graphic toolkit [28] have been compiled and installed (vers. 3.0.0 for 64bit O.S.) on the remote HPC infrastructure. In this framework the object geometry is stored in the so called "Drawable" object. At every video frame, OpenSceneGraph determines what "Drawable" object is visible (culling algorithm) and in the following "draw" process the "Drawbles" objects will be rendered according with the distance from the eye point [29]. The application runs on the remote machine (server) using the hardware resources including the GPU for rendering. The images, according to the user requests, are rendered, compressed and streamed via web to the client which have only to decompress and visualized them in the viewer (Fig. 4). The 3D model which can be remotely visualized are all the file formats supported by OSGViewer.

### *B. Editing mode*

The remote display can be basically done through any graphic application, both proprietary or open-source, using OpenGL technology on a Linux operating system (O.S.). In order to test the possibility of processing and editing 3D data, another solution available in ARK3D for the remote rendering of large 3D models is given using Meshlab [30]. The program, installed at 64 bit on Linux Centos O.S., is called and 3D models (all those supported) can be displayed and edited using the function available in the open-source program (Fig.5). Thus the user can load and work with range maps, unstructured point clouds or polygonal data (millions of triangles), even on standard laptop with limited computing resources, edit and analyse the model, extract geometrical information or run batch processes. This kind of approach is intended to stimulate sharing and diffusion of 3D models among the communities in order to promote collaborative works among technicians and non-experts without limitations imposed by missing computing resources.



a)

b)

Figure 5: Example of a UAV-derived cloud with 108 mil. points with related RGB colour loaded remotely using Meshlab (a). A textured polygonal model with 3 mil. polygons and ca 115 MB texture displayed within Meshlab (b).

| *Examples* | *Tomb 1 – fig. 4* | *UAV cloud – fig.5a* | *Tomb2 – fig.5b* | *Mountain – fig. 6* | *Temple – fig. 7* |
|---|---|---|---|---|---|
| Geometry | 410K polygons 204K vertices | 108 mil. points | 3 mil. polygons 1.5 mil vertices | 1.5 mil. polygons 470K vertices | 40 mil. polygons |
| RGB Data | 2 GB TIF Texture | Colour per point | 115 MB JPG Texture | 100 MB JPG Texture | - |
| File Format | OBJ | ASCII PLY | OBJ | OBJ | PLY |
| File Dimension | 43 MB | 1.6 GB | 345 MB | 200 MB | 878 MB |

TABLE 1: CHARACTERISTICS OF THE DATASETS USED TO TEST THE REMOTE RENDERING PLATFORM.

Figure 6: Example of large landscape 3D model – the Three Picks in Lavaredo, Dolomites. The geometry counts ca 1.5 mil. polygons with texture.

## VI. CONCLUSIONS

The paper presented a new ICT methodology for remote rendering of large 3D models, in form of structured (i.e. mesh) or unstructured (i.e. point cloud) 3D data. The method, developed at the ENEA research centre, is based on a hardware-software / client-server architecture which exploits the great performances of a HPC infrastructure included in the first 500 supercomputers worldwide.

Different models were used for the testing, including textured polygonal models as well as large coloured point clouds. The interactive remote renderings were performed with PCs or normal laptops, on LAN or Wi-Fi Internet connections, showing very good performances and smooth visualization with both viewers (OSG and Meshlab).

With respect to other approaches, the presented method (i) is multi-user, (ii) frees the user from the need of specific hardware and software resources, (ii) protects the copyrights related to the 3D applications and data, since the model won't be downloaded locally, (iii) allow some processing/editing on the remote 3D data and (iv) exploit the potentialities of an HPC infrastructure on the server-side. The user interacts with the virtual environment using remote hardware and software resources and receive, locally, only uncompressed images which simulate the interaction with a real 3D model. The initial waiting time for the remote HPC computer to load the model is generally quite short, even for the largest models, always depending on the available network connection.

This kind of 3D data visualization and sharing approach is intended to stimulate a diffusion of 3D models among the communities. Technicians can more easily share and transfer information with the other communities (e.g. heritage) with no limitations imposed by high computing resources or violation of IPR.

## VII. REFERENCES


[1] A.M. Manferdini, F. Remondino, "A review of reality-based 3D model generation, segmentation and web-based visualization methods", in Int. Journal of Heritage in the Digital Era, Vol.1(1), pp. 103-124, 2012.

[2] D. Koller, M. Tutitzin, M. Levoy, M. Tarini, G. Croccia, M. Cignoni and R. Scopigno, "Protected interactive 3D graphics via remote rendering", in Proceedings of ACM SIGGRAPH, pp. 695-703, 2004.

[3] F. Uccheddu, M. Corsini, M. Barni, "Wavelet-based blind watermarking of 3D models", in Proc. ACM Multimedia and Security Workshop, pp. 143-154, 2004.

[4] J. Zhu, J.Z. Bakdash, D. Koller, T. Banton, D.R. Proffitt, and G. Humphreys, "Quantifying usability in secure graphics: assessing the user costs of protecting 3D content", in Proc. of the Symposium on applied Perception in Graphics and Visualization (APGV), 2008.

[5] E. Zagrouba, S.B. Jabra, "A new approach of mesh watermarking Based on maximally stable meshes detection", in Proceedings of the 3rd International Conference on New Technologies, Mobility and Security (NTMS), pp.1-5, 2009.

[6] D. Abate, G. Furini, S.Migliori, S. Pierattini, "Multiple visualization web approach for Cultural Heritage objects" in Journal Geoinformatics FCE CTU, Volume 7, Dec. 2011.

[7] E. Gobbetti, F, Marton, P. Cignoni, M. Di Benedetto, F. Ganovelli, "C-BDAM - Compressed Batched Dynamic Adaptive Meshes for terrain rendering" in Computer Graphics Forum, Vol. 25(3), pp. 333-342, 2006.

[8] C. Andujar, P. Brunet, "Relief impostor selection for large scale urban rendering", In IEEE Virtual Reality Workshop on Virtual Citiscapes: Key Research Issues in Modeling Large-Scale Immersive Urban Environments, 2008.

[9] S. Yoon, E. Gobbetti, D. Kasik, D. Manocha "Real-time massive model rendering", in Synthesis Lectures on Computer Graphics and Animation (vol. 2), 2008.

[10] M. Di Benedetto, P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, R. Scopigno, "Interactive remote exploration of massive cityscape", in Proc. of the 10th VAST Conference, pp. 9-16, 2009.

[11] C, Andujar, J. Boo, P. Brunet, M. Fairén, I. Navazo, P. Vázquez, À, Vinacua, "Omni-directional relief impostors" in Computer Graphics Forum 26(3), pp. 553–560, 2007.

[12] H. Buchholz, J. Döllner "View-dependent rendering of multiresolution texture-atlases", in IEEE Visualization, p. 215–222, 2005.

[13] P. Cignoni, M. D. Benedetto, F. Ganovelli, E. Gobbetti, F. Marton, R. Scopigno, "Ray-casted blockmaps for large urban visualization", in Computer Graphics Forum 26(3), 2007.

[14] http://graphics.stanford.edu/software/scanview/

[15] D. Koller, B. Frischer, "The Digital Sculpture Project: Casting light on ancient sculpture with new 3D technologies", in Proc. CAA, 2010.

[16] http://www.ccrmlabs.com/

[17] S. Rusinkiewicz, M. Levoy, "QSplat. A multiresolution point rendering system for large meshes", in Proc. 27th annual conference on computer graphics and interactive techniques, pp. 343-352, 2000.

[18] E. Gobbetti and F. Marton, "Layered point clouds: A simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models", in Computer and Graphics, 28(6), 2004.

[19] S. Rusinkiewicz, M. Levoy, "Streaming QSplat: a viewer for networked visualization of large, dense models", in Proceedings of the. Interactive 3D Graphics (I3D) Symposium, 2001.

[20] A.S. Hussein, S.H. Hamad, A.H. Abdelaziz, S.H. Abdelaziz, H. El-Bery, "Parallel remote rendering of large 3D point-Based models on mobile clients", in Proc. CIMSiM, pp. 419 – 426, 2010.

[21] P. Cignoni, F. Gavonelli, E. Gobetti, F. Marton, F. Ponchio, and R. Scopigno, “Adaptive TetraPuzzles – efficent out of core construction and visualization of gigantic polygonal models”, in ACM Transaction on Graphics (Proceedings of ACM SIGGRAPH 2004), 23(3), pp. 796-803.
[22] Y. Okamoto, T. Oishi, K. Ikeuchi, “Image-based network rendering system for large sized meshes”, Proc. ICCV, pp. 931-938, 2009.
[23] Y. Okamoto, T. Oishi, K. Ikeuchi, “Image-based network rendering of large meshes for cloud computing”, in International Journal of Computer Vision, (94), pp. 12-22, 2011.
[24] B. Frischer, “New directions for cultural virtual reality: a global strategy for archiving, serving and exhibiting 3D computer models of cultural heritage sites” in Proceedings of Virtual Retrospect Conference, pp. 168-175, 2006.
[25] D. Koller, B. Frischer, G. Humphreys, “Research challenges for digital archives of 3D cultural heritage models”, in Journal on Computing and Cultural Heritage (JOCCH), Vol. 2(3), doi 10.1145/1658346.1658347, 2009.
[26] https://www.ark3d.enea.it/home.php
[27] http://www.top500.org/
[28] http://www.openscenegraph.org
[29] T. Moore, “Open Scene Graph States and StateSets”, in http://www.bricoworks.com/articles/stateset/stateset.html, 2010.
[30] http://meshlab.sourceforge.net/
[31] F. Fiorillo, B. Jiménez Fernández-Palacios, F. Remondino, S. Barba, “3D Surveying and modeling of the archaeological area of Paestum, Italy”, in Proc. Arqueologica 2.0, Sevilla, Spain.

Figure 7: A large polygonal model of the Neptune temple in Paestum [31] with ca 40 mil. Polygons.