CREATING, VISUALIZING AND SHARING HIGH-POLY AND MULTI-TEXTURES MODELS ON THE WEB

F. Rechichi[1], F. Fiorillo[1]

[1]Department of Architecture, Built environment and Construction engineering, Politecnico di Milano, Italy
(fabrizio.rechichi, fausta.fiorillo)@polimi.it

Commission II

KEY WORDS: Texture Mapping, High-resolution Mesh, Web Sharing, WegGL, Web Information System

ABSTRACT:
In many Cultural Heritage cases study, where usually the shapes to be reproduced have complex and irregular geometries there is a growing demand to create and use very high-resolution polygonal models that represent the real objects with great accuracy and level of detail. In archaeological field, for example, it is fundamental to create a virtual reconstruction that is as close as possible to the reality, because the digging operations are often destructive and it is necessary to track and preserve for the future as much as possible. This requirement leads to three relevant problems: 1) the elaboration of high-poly and high-resolution models, 2) the management of these models, 3) their sharing and access. In this paper, both a scientific approach for realizing the accurate and detailed models and a system to manage, share and visualize these models on the web will be shown. Moreover, a multi-texture digital model elaboration is proposed for the correct definition of the geometrical and texture resolution in relation to the survey level of detail, the physical size of the object and the project requirement. In particular, a web-system that allows the sharing of very high-resolution models, with multi-textures support will be presented.

1. INTRODUCTION

Nowadays there is always more need to create and use very high-resolution polygonal models that represent the real objects with great accuracy and level of detail. Moreover, this need is really a demand in many Cultural Heritage’s cases of study and a specific request in the archaeological area, where usually the shapes to be reproduced have complex and irregular geometries. In archaeological field, for example, it is fundamental to create a virtual reconstruction that is as close as possible to the reality, because the digging operations are often destructive and it is necessary to track and preserve for the future as much as possible. This requirement leads to three relevant problems: 1) the elaboration of high-poly and high-resolution models, 2) the management of these models, 3) their sharing and fruition.
In this paper, both a scientific approach for realizing accurate and detailed models and a system to manage, share and visualize these models on the web will be shown.

1.1 State of art

The interest in digital models and 3D reconstructions invaded many disciplines, even different among them, of engineering (industrial, mechanical, civil-environmental but also medical), architecture, design and Cultural Heritage (CH); consequently, it also involved very different range of acquisition and dimensional scales (from the detail of a vase or a mechanical piece of modest size compared to the landscape reconstruction).
The widespread use of reality-based 3D models in CH field (Gonzalez-Aguilera, D et al., 2018) has certainly been facilitated by the unstoppable progress of technological research, related to: i) hardware (active and passive sensors), ii) software (above all for the management of large point clouds and big data) and iii) algorithms, especially those for accelerating and optimizing the photogrammetric process (Fassi, et al. 2017; Remondino et al, 2017).
This aspect was followed by the growing need to publish online the captured 3D data, in order to have the possibility to share, visualize, measure, edit and co-work in real time. This demand has also been demonstrated by the increasing diffusion of platforms or web-databases created ad hoc. Some of these are produced by software and hardware house, e.g.: the FARO “SCENE WebShare” package (FARO Technologies, 2018) and the “TrueView” from Leica Geosystem. These two applications are a data-based solutions for mainly storing, sharing, measuring and marking up scan data (point cloud) on internet with different project working partners or even with no experts in the field of 3D surveying. Instead, one of the most widespread and popular platforms where it is possible to publish textured polygonal models is definitely Sketchfab[1].
In addition, there are numerous research projects in the field of valorization and fruition of CH, that prove the growing interest into the web-sharing of 3D reconstructions for: i) dissemination purposes aimed at a non-expert public, or; ii) technical multi-disciplinary and interdisciplinary working aimed at professional experts (structural engineers, archeologists, restorers, architects, etc.). Some examples are: i) MayaArch3D[2] “a virtual research environment for the documentation and analysis of complex archaeological sites – specifically, it is a web-based 3D-GIS, that can integrate 3D models of cities, landscapes, and objects with associated, geo-referenced archaeological data.”; ii) Scan4DReco[3] “Multimodal Scanning of CH Assets for their multilayered digitization and preventive conservation via spatiotemporal 4D Reconstruction and 3D Printing”; iii) DigiArt[4] “a solution to the capture, processing and display of cultural artefacts”.
Definitely one of the most complete and famous multi-function platforms appears to be 3D HOP (Visual Computing Laboratory, 2018): an open-source framework for the creation of webpage to share polygonal 3D models, from where users can extract metric and textual information. It is used by the European platform ARIADNE (a large online database of finds coming mainly from European archaeological sites) to publish 3D models. 3D HOP represents an excellent tool, a useful showcase and a significant boost to promote the adoption of 3D reconstruction in the field of CH and to show its potential.

[1] https://sketchfab.com/feed
[2] http://www.mayaarch3d.org/language/en/sample-page/
[3] http://www.scan4reco.eu/
[4] http://digiart-project.eu/

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-W9-655-2019 | © Authors 2019. CC BY 4.0 License.

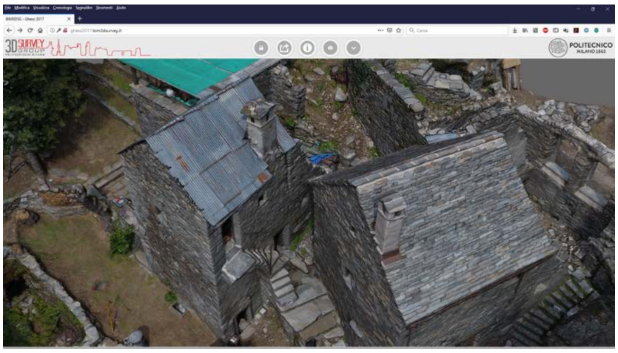


Figure 1: Textured polygonal model of the Ghesc’s site on BIM3DSG

For the presented research work, the BIM3DSG system was used (Fassi et al., 2012 and Rechichi et al., 2016); it is an online and ad hoc data-based web-platform capable of managing simultaneously point cloud, textured high-poly models, orthoimages, photos, textual information, and technical documentations. This system solution has been designed for facilitate the communication and interoperability between different professional and no-skilled users, in an online web-system; one of its strengths is the capability to manage high-poly multi-texture mesh model. This aspect is powerful in the world of CH. where the colour representation is essential.

2. CHOOSE THE RIGHT TEXTURE RESOLUTION

Even if the power of computers has grown considerably in the last years, it is not infinite. For this reason, it is fundamental to compute rightly the resolution of the model and of the texture, in precise relation to the resolution of the survey, the physical dimensions of the 3D reconstruction and to the project requirements, in order to not waste valuable resources.

It is important to remember that the uncompressed size of a typical single texture (24bit, 8bit for 3 colour channels) of 4.096x4.096 pixel is 48MB (4.096x4.096x3Byte[5]). All textures must be loaded into the Video RAM, which must also dedicate some space for the render of the geometry: when the Video RAM is full, the Video Card can’t draw anymore the scene. Modern Video Cards automatically adopt some optimizations and compression at runtime, that let to reduce the uncompressed texture size even by 70-80%; even with this high but unpredictable reduction, the Video RAM set up a real limit to the max textures that can be loaded: for example, with an 8GB Video Card the limit is around 10-14Gpixel[6], that are around 600-850 textures of 4.096x4.096 pixel.

Furthermore, to ensure a good compatibility with the most commonly used software, it is better that a single texture is maximum 4.096x4.096 pixel, because not all of them support textures of higher resolution and many can’t rightly manage textures beyond 8.192x8.192 pixel.

These requisites lead to the necessity to use the multi-textures technique for objects of which the dimensions or the quality of the survey would need a texture’s size higher of 4.096x4.096 pixel. When multi-textures are building, it is better to choose a fixed single texture size of 4.096x4.096 pixel, adapting only the textures count, because if it would be bigger compatibility problems may occur and if it would be smaller unnecessary overhead arise.

Otherwise, if the object is very small or the resolution of the survey is not enough to really require a texture of 4.096x4.096 pixel, it is important not to waste precious resources and to choose a smaller texture size for that model, remembering that it should be squares with sides power of two, for technical informatics reasons, thus choosing between 2.048x2.048, 1.024x1.024, 512x512, 256x256, 128x128, 64x64,32x32, 16x16, 8x8, 4x4, 2x2, 1x1 pixel.

It is possible to calculate the right dimension of textures, starting from the area of the model surface and its geometric resolution. In fact, the number of textures can be determined simply by dividing the mesh area by the square of the GSD (Ground Sample Distance) and by the resolution (in pixels) chosen for the single image. From the same equation, fixing the count to 1, it is possible to calculate le resolution (in pixel) for objects that require smaller textures,

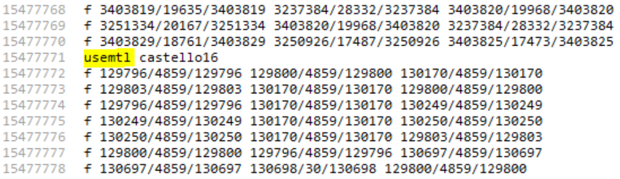
2.1 Multi-texture

The internal software management of multi-textures models is conceptually simple: the model is subdivided in a number of parts equal to the chosen number of textures, each of them having a single texture of the chosen size; all this “sub-models” are managed as a whole, as if they were a single mesh model.

This operation is completely transparent to the user, that see and can interact with only a single model, even if it is composed by many parts.

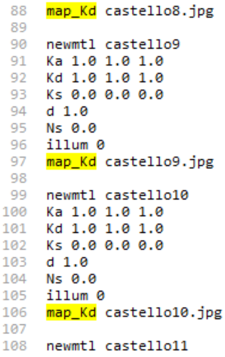
The exported OBJ of a multi-textures model also reflects this particular internal substructure: the vertex (v), the vertex normal (vn) and the vertex’s textures coordinates (vt) are managed all together, while the faces (f), the triangles composing the mesh, are subdivided into section using the usemtl directive: each sections correspond to a “sub-model” and to a single texture.

The relationship between these sections and the corresponding textures can be found into the mtl file, referring to the newmtl directive, that describes separately for each part the material and the name of the texture file, using the map_Kd directive.



```
f 3403819/19635/3403819 3237384/28332/3237384 3403820/19968/3403820
f 3251334/20167/3251334 3403820/19968/3403820 3237384/28332/3237384
f 3403829/18761/3403829 3250926/17487/3250926 3403825/17473/3403825
usemtl castello16
f 129796/4859/129796 129800/4859/129800 130170/4859/130170
f 129803/4859/129803 130170/4859/130170 129800/4859/129800
f 129796/4859/129796 130170/4859/130170 130249/4859/130249
f 130249/4859/130249 130170/4859/130170 130250/4859/130250
f 130250/4859/130250 130170/4859/130170 129803/4859/129803
f 129800/4859/129800 129796/4859/129796 130697/4859/130697
f 130697/4859/130697 130698/30/130698 129800/4859/129800
```

Figure 2: Part of a multi-textures OBJ file showing faces sections using usemtl directive



```
map_Kd castello8.jpg

newmtl castello9
Ka 1.0 1.0 1.0
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
d 1.0
Ns 0.0
illum 0
map_Kd castello9.jpg

newmtl castello10
Ka 1.0 1.0 1.0
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
d 1.0
Ns 0.0
illum 0
map_Kd castello10.jpg

newmtl castello11
```

Figure 3: Part of multi-texture mtl file showing mtl section and texture references with map_Kd directive

[5] 8 bit = 1 Byte.

[6] It is no possible to precise calculate it because the real amount of reduction due to the Video Card’s optimizations and compression varies from texture to texture.

3. ELABORATION PIPELINE

A close-range photogrammetric survey proved to be the most appropriate approach to the tested research works because it responds at the same time to the need of high mesh definition and high texture resolution model.

For the tested digital models presented in this paper, the standard image-based modelling pipeline was used: i) photos alignment (self-calibration and external orientation); ii) marker detection for the camera calibration optimization; iii) scale and eventually geo-referencing of 3D reconstruction using target distance measurement (for small object) or a topographic network (for architectural and landscape representation scale); iv) dense cloud and mesh model production; v) polygonal model texturing.

Before the last texture-mapping step, it is important to perform an additional editing phase to optimize the polygon model surface. In fact, the mesh generated by a point cloud model, are often subject to morphological and topological errors, which must be corrected to improve and regularize the polygonal surface (Verhoeven, 2017).

The most common errors are: i) triangles with an orientation incoherent compared to the normal of adjacent polygons (dihedral angle greater than 90°); ii) triangles with an inner angle less than 10-20° (in fact, the equilateral triangles optimize the quality of the surfaces); iii) degenerate triangles that have twice the same vertex (spikes); iv) overlapping and coincident triangles; v) isolated groups of triangles disconnected, or connected with a single vertex; vi) edges shared by two triangles; vii) vertices shared by multiple disjoint triangles; viii) intersecting triangles (Bolognesi et al, 2018).

In the presence of possible gaps on small areas of the model, due to undercuts or to problems of survey, it is necessary to operate with surface reconstruction algorithms. "Filling" generates new surfaces depending on the curvature of the edges, on the tangents to adjacent polygons or simply flat.

Finally, other algorithms are also available with the aim to smooth and regularize the surface and/or to decimate the number of polygons. The latter operation is often indispensable for the publication on same web-platform or for exporting the final digital model in other three-dimensional modelling programs. BIM3DSG has no problem of management of high-poly mesh, thus for the case studies presented this last step was not performed. However (as better specified below) using a multi-texture approach, it can sometimes be convenient to decimate the models and let some details be represented by texture that in the photogrammetry has a metric consistence.

For the next texture mapping process, it is better if the final polygonal model satisfies some geometrical requirements: it must be artefact-free, hole-free, and two-manifold.

In summary, the main steps related to the mesh surface editing (reverse modelling phase) generally are: mesh repair, filling holes, decimation (eventually), and surface optimization. Once obtained a digital model optimized and without morphological and topological errors, it is possible to take care of the texture mapping operation to associate colour to the geometric information (without increasing the geometric complexity).

In general, there are two main types of texture: repeatable textures and specific textures strictly connected to the geometry of mesh model.

The first type has the advantage of being able to be repeated endlessly on a surface, without any problem of resolution. However, can only be used in the case of digital reproductions of surfaces without morphological singularities and with a total geometric regularity. In fact, it is widely used in the field of entertainment and video games but not very useful for CH purposes.

The second type has a very close relationship with the geometry for which it is generated: each portion of the texture must be closely related to the corresponding detail of the polygonal model. In this case, the resolution of the image to be projected, that is the pixel size, will influence the visible geometric detail and must therefore be proportional to the real physical dimensions of the digital reconstruction.

The key factor of the image-modelling process is the exact correspondence between the geometric mesh detail and its RGB color data, that cannot be ensured with other texture mapping techniques (procedural texture mapping, repeated/mirrored texture mapping, planar/cylindrical/spherical texture projection with homologous point, directly painting onto 3D surfaces, etc.). In addition, this direct connection pixel/geometry, also allows determining the correct resolution of the texture to be mapped on the surface, related to the physical size of the object.

According to the considerations presented in paragraph 2, it is better to produce 4096x4096 pixels texture in order to allow compatibility with the majority of the visualization and management systems of polygonal models. Consequently, it is necessary to create a multi-texture digital model dividing the image to be projected onto the surface in 4096x4096 pixel tiles and determining their correct number according to the physical size of the object and the geometric resolution (GSD). Once determinate the area of the mesh, it is divided by the square of the GSD and the resolution of 4096x4096 pixels to obtain the number of tiles.

Technical information about survey and its elaboration of Umm al-Dabadib Fort case study are reported below. The Fort (Figure 4), located in the middle of the Fortified Settlement (Kharga Oasis – Egypt), consists of a square building (27x27 m) of 5 floor levels (about 13 m) with two rectangular towers on south side. In order to survey the exterior, 509 photos were acquired using a Canon EOS 5D Mark III with a fix 35mm lens. The average capture distance from the object was of about 15-20m; related to the camera sensor resolution, this distance ensured an average GSD of 1,8 mm. So being the mesh area of 1420 m^2, it was possible to generate 25 textures of 4096x4096 pixels.

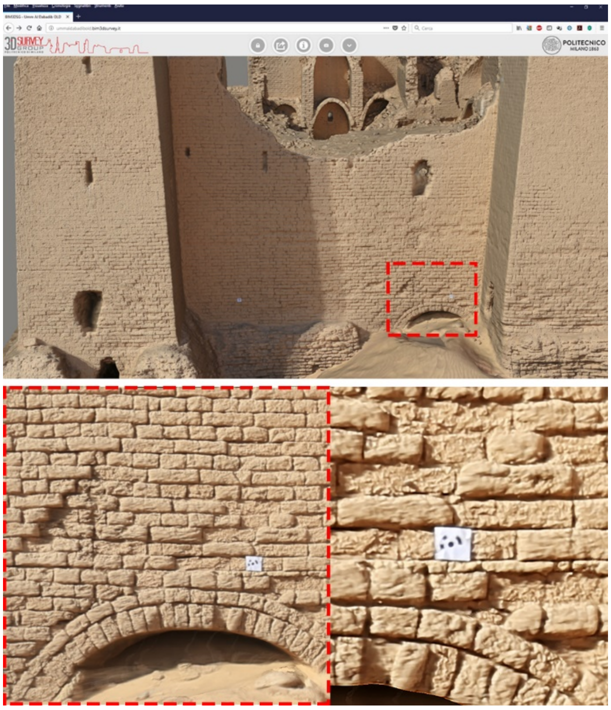


Figure 4: Polygonal model of Umm al-Dabadib Fort on the BIM3DSG system, composed by 25-texture 4.96x4.096 pixel (area mesh 1420 m^2, GSD~2mm)

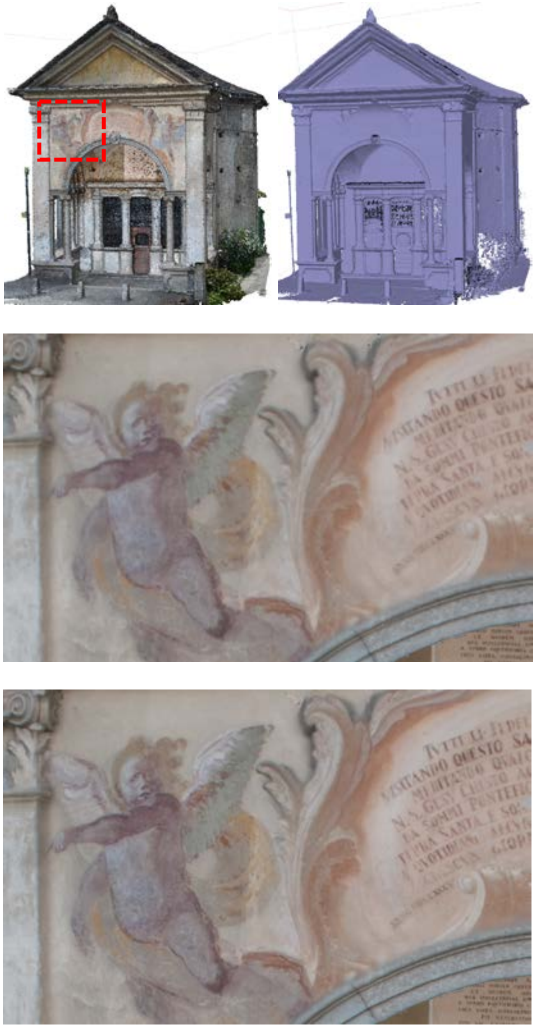


Figure 5: 3D model of Chapel 2 (Sacro Monte del Calvario di Domodossola – Italy) on the top. The details show the difference between a model with 1 texture (center) and 8 textures of 4096x4096 pixels (bottom)

In Figure 5 the image-based model of the Chapel 2 of the Sacro Monte del Calvario di Domodossola (Italy), a UNESCO complex of a sanctuary and fifteen chapels representing the stations of "via crucis", is shown.
The 3D reconstruction is obtained using 400 photos (Nikon D3000 -18 mm focal length); the average GSD is 2 mm and the mesh area is about 526 m^2. The detail zoom in the lower part, shows the difference between the same geometric model (3 million polygons) mapped with 1 texture (center) or 8 textures of 4096x4096 pixels (bottom).

4. MANAGE, SHARING AND ACCESS

The web-system presented here allows the management, the sharing and the fruition of very high-resolution models, with multi-textures support. It is a specific extension of the BIM3DSG system, presented in Fassi at al. (2015) and in Rechichi at al. (2016), covered by Italian Patent Pending (2014).
The system is based on two different components:

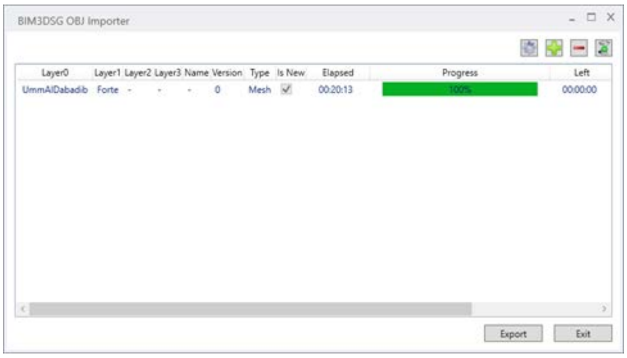
1. a stand-alone OBJ Importer, that allows to import into the system OBJ models, with multi-textures support;
2. the web interface, that allows users to navigate the models on the web, besides giving some additional features, such as a very complete information system, hotspots and orthophotos supports, measurement tools, …

The core of the entire system is the database, a PostgreSQL, that contains not only the information, but also all the models, the textures and the files, excepts for the orthophotos that are stored directly in the filesystem. Having a central database that includes all the data lets to have concurrently access from different sites around the world and facilitates migration and backup processes. It is chosen to use a web interface, instead a standalone software, because it helps the diffusion, the fruition of the system and the dissemination of the data: in fact every user already owns a web browser on his device and has great familiarity with it; furthermore, it is cross platform and doesn't require any additional effort both from the end user and the organization which want to share the data.

4.1 The OBJ Importer

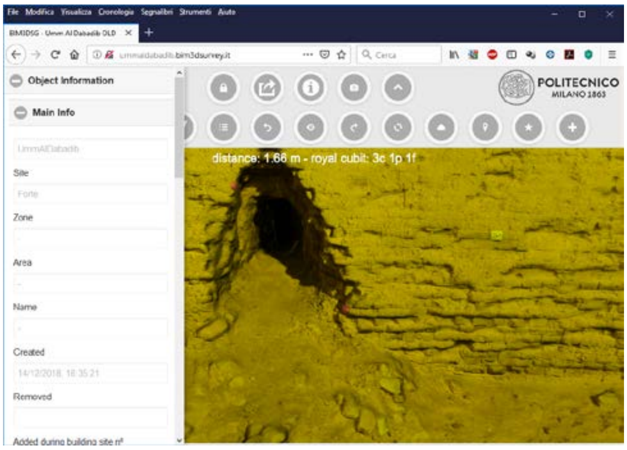
The OBJ Importer is a standalone software that allows to import OBJ models into the system, without requiring the user to buy or use any external software. Furthermore, it lets the user to use his favorite software to produce the models, because OBJ file is a widely supported standard and every model can be easily saved or converted into an OBJ file.
The tool is written from scratch in C# and WPF, with the addition of ITinnovations Libraries (Rechichi F., 2018), and at the moment it is available only from Windows.
The OBJ Importer supports both un-textured, textured models and also multi-textures; it automatically recognized the textures reading the mtl file.
The OBJ Importer currently asks the user to respect some easy constrains for the OBJ file to be input, that are available as save option in most used software: it must contain only one model, described with both the (v) and the (vn) and the (vt), besides the (f), composed by only triangles[7], and each entry must be inline.
The OBJ Importer automatically computes seven lower levels of detail (LoD), useful to see the model on low level devices; this computation is done through some MeshLabServer's scripts, that use "Simplification: Quadric Edge Collapse Decimation" filter with a target percentage reduction of respectively 0.9, 0.8, 0.6, 0.4, 0.2, 0.1, 0.05 percent; each filter is always executed on the original file, to avoid cascade error propagation.
The original model and each lower LoD are converted into JSON files and stored into the database; the original OBJ are also stored into the database, to lets user to download it. If there is a multi-textures model, it is at first subdivided into sub-models, one for each texture files. Due to the limitation of JSON files, if each model or sub-model has more than 65.536 vertex or faces, the OBJ Importer must firstly subdivide it into smaller separated models, that are then managed by the system as a whole.
The OBJ Importer is specifically written with great parallelism and many optimizations to lower the computational cyclomatic complexity, at the expense of around the double needed memory used. Thanks to these optimizations, it has very low importing times, that principally depends on the time used by MeshLabServer to compute the lower LoDs; the upload is executed in background and, if it is present a good internet connection, it doesn't influence much the time, but can be an important bottleneck in case of poor upload bandwidth, such as on home ADSL.

[7] The mesh containing squares can be easily converted in mesh with only triangles, through an automatic function that uses two triangles for each square.



Layer0	Layer1	Layer2	Layer3	Name	Version	Type	Is New	Elapsed	Progress	Left
UmmAlDabadib	Forte	-	-	-	0	Mesh	☑	00:20:13	100%	00:00:00

Figure 6: BIM3DSG OBJ Importer at works


File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto
BIM3DSG - Umm Al Dabadib OLD
ummaldabadib.bim3dsurvey.it
Cerca
POLITECNICO MILANO 1863
distance: 1.66 m - royal cubit: 3c 1p 1f

Object Information

Main Info

UmmAlDabadib

Site

Forte

Zone

Area

Name

Created

14/12/2018, 18:35:21

Removed

Added during building site n°

Figure 7: Example of some additional function of BIM3DSG

4.2 The viewer

The 3D web viewer allows to load concurrently any number of models and textures; the only limits are due to the availability of System RAM and Video RAM. The System RAM set up a limit for the combination of JSON files, used by BIM3DSG for the 3D models, and texture files: they must be all in RAM, otherwise for drawing each frame the computer should be read it from the disk, that it is no possible with the current technology, because even solid-state disks are to slow to support an adequate frame rate. The Video RAM limits are already described in detail in the second chapter.

To ensure to have always the best possible quality, it is chosen to not adopt dynamic resolution changes in the model, as most online viewer done, but to act in the same way of any modelling software, thus loading always all the model at the full resolution; however, the user has the possibility, both at loading time or selecting an object, to choose to use the original quality or one of the lower LoDs, both for the models and the textures, to adapt the scene to the power of the used device.

The web viewer is based on WebGL (Di Benedetto et al., 2014), through the SceneJS library (SceneJS, 2018). To ensure the maximum interoperability and compatibility, the user interfaces is built using only HTML5, CSS5, PHP, JavaScript and JQuery, that are todays supported by most of browsers and OSes, both for workstations and mobile devices.

To lower the loading times, considering the use of huge high-resolution models, it is written an ad-hoc cache system, that not depends from the usual browser cache, because there is the need to directly control it, to make it permanent and to use more space than which usually assigned for it. BIM3DSG uses the persistent storage (Web Persistent Storage, 2018) with three different non-relational databases, used for the cache of the: i) models, ii) textures, iii) images and the other files.

The viewer offers complete controls to navigate, rotate and zoom, both with mouse, keyboard and also touch, to ensure the best usability with every device.

The viewer also supports some advanced functions, such as the capability to take some measurement, and it is also possible to take the high-resolution images of the current scene.

Clicking on one object, with mouse or touch, it is selected and it is possible to performs many actions on it, such as i) access to automatically computed information, ii) visualization or modification of custom information, images, orthophotos and files associated with that object, iii) addition of maintenance events and iv) visualization or modification of information, images and files associated with each maintenance event.

Finally, there is also the possibility to insert and view hotspots, that are treated in the same way as other objects, therefore with the support to custom information, images and files and also to maintenance events. Hotspots are very useful both to set up information points or to insert objects, such as archeological finds, while waiting to create their model or for which you do not want to create a model.

5. PERFORMANCE ANALYSIS

When it is the need to work with very high resolution and/or very huge models, the performance and the scalability of the system became a key aspect to ensure a good usability and the fruition of the system.

These aspects became even more important when the models must be used by not 3D specialist and for the dissemination. Common users don’t have or don’t want to buy super computers or high end workstation to use the system; furthermore they don’t keen of waiting for long loading times or seeing stuttering or having poor frame rate and they quickly give up a system characterized by poor user experience.

5.1 Hardware used for tests

Where not differently specified, the tests are run using a good PC, but not a workstation or a top PC; the two years old PC is an Intel i7-7700K, with 64GB of system RAM and a Nvidia GTX1070 with 8GB of Video RAM by ASUS. The models and the browser cache are both placed on a Samsung 960 PRO SSD, to minimize the impact of the disk’s speed.

The browser used for the test is Firefox 64.0, 64bit version. The other software’s versions are: MeshLab 2016.12, 3DHOP 4.2, Microsoft 3D Builder, Agisoft 1.4.3, Rhinoceros 5.14, Geomagic Freeform Plus 2019.0.61.

To test the scalability of BIM3DSG, it is used both on a low speed tablet and an old middle range smartphone, respectively a very cheap five years old Windows tablet by ASUS, with Atom 32bit, 2GB RAM, no external Video Card, and a “five editions old” smartphone, a Samsumg A5 2015, with 1,5GB RAM, connected in 4G to internet.

5.2 Methodology

To test the loading times, the models is loaded five times, then the lower and the higher times are discarded and finally, it is taken the average of remaining values. After each load, the software is closed and then open again to avoid internal optimizations and to start each time from the same conditions. To test the frames per second (FPS)[8], it is used FRAPS, a tool that

[8] The FPS is the way to measure performances in the computer graphic area; it is important both the stability of the FPS and its value: a value of 60 FPS is perfect, between 30 and 60 FPS is still acceptable for the fruition, but a good eye sees less

lets to show and record the FPS from DirectX and OpenGL applications; it is recorded 20 seconds of navigation of the model, rotating, panning and zooming it in and out.They are reported the values saved by FRAPS; it is important to remark that the frequency of the used full HD monitor is 60Hz, thus the max FPS are locked at 60fps. For all software are used the standard video settings.
After testing each software and after the load test, the computer is rebooted to have the same conditions in all tests.

5.3 Multi-texture Model and performance comparison

5.3.1 The model

For this test it is used a multi-textures model of the Forte of Umm al-Dabadib in Egypt. The model is composed by 6.803.780 faces (triangles) and 3.406.360 vertex; the texture is composed by 25 JPEG files of 4.096x4.096 pixel each, thus having a total texture size of 419.430.400 pixel (419,43Mpixel).
The size of the OBJ file is 858MB, plus 56,7MB of JPEG image's files for textures (Figure 8).

5.3.2 BIM3DSG

The loading time of the model when it is already in the cache are the best in this test: BIM3DSG takes only around 8,01 seconds to draw the scene. The first time it is loaded, the time instead depends principally from the internet connections: using a 100Mbit/s bandwidth it takes around 33,81 to download all the data from internet and render the scene.
The FPS is fully stable at 60fps, that is the monitor locked refresh of the PC used, so it could be even higher on a higher refresh monitor. BIM3DSG shows not only the best FPS in this test, but also a perfect experience and usability.
The max GPU usage is under 25%, so there is a lot of computational resources both to load together many other models and/or to use higher refresh rates, needful for example in virtual reality applications. The RAM used by the system is only 1,5GB.

5.3.3 Loading time comparison

The loading times (Figure 9) show that BIM3DSG is the best, closed followed by 3DHOP, in ply mode, but 3DHOP doesn't load the textures and seems not to support multi-textures. Agisoft is at third position, despite it must load all the project and not only the model. Rhinoceros is the software that takes more time to load the OBJ and render it, but its primary role isn't to work with obj, mesh and textures.

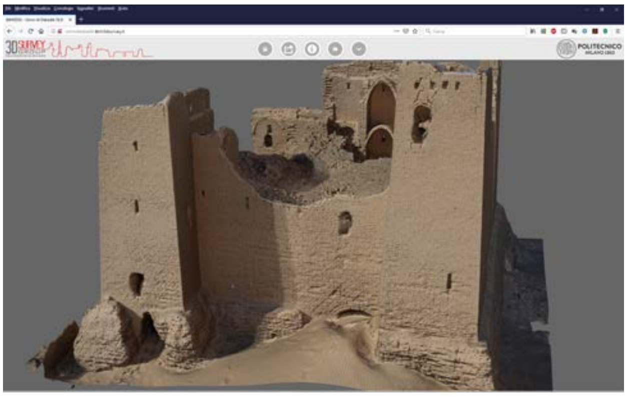


Figure 8: Forte of Umm al-Dabadib in Egypt on BIM3DSG

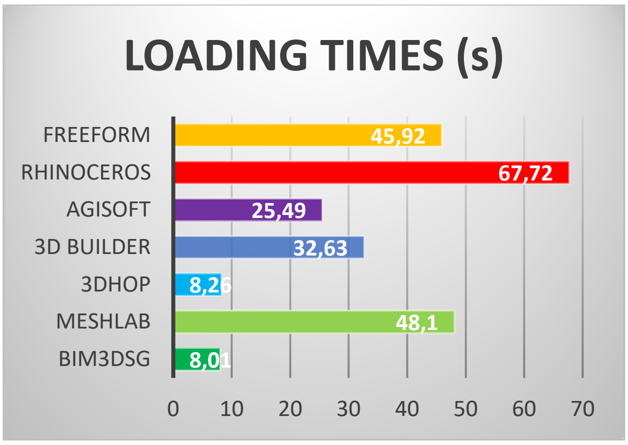


Figure 9: Loading of Umm al-Dabadib's Forte; lower is better

Agisoft is then set as the reference (100%) of loading times (Figure 10), because it is the software with which the model is created. BIM3DSG takes only the 31,42%, thus is 650% better respect Agisoft, 573% better respect Geomagic Freeform Plus and 845% better respect Rinocheros.

5.3.4 FPS comparison

In the FPS test (Figure 11), BIM3DSG outperform every other tested software: not only it has the best value, but its frame rate is perfect stable and it is limited by the monitor locked refresh, indicating that it can manage higher FPS without any problem.
3DHOP's FPS for only the model, because it doesn't load multi-textures, is very unstable, especially during the zoom phases; even if the average value is acceptable, the many drops down lead to a very poor experience. Furthermore, it uses around 40% of GPU and few CPU, despite it doesn't perform well, thus it isn't able to use all the available PC resources.
At the third place we have Microsoft 3D Builder, even if it is not possible to precise track the fps inside 3D Builder because it is incompatible with FRAPS and doesn't offer an internal tool to show the fps: it looks around 30/40fps, but with many drops under the 30fps and many stuttering effects. Despite Microsoft 3D Builder shows quite good performance, it is important to underline that 3D Builder seems to adopt some simplifications and speed optimizations at the expense of image quality: it shows a very different look from all the other software tested here and it doesn't manage the source light and the shadows in a right way.
The Freeform's frame rate is quite stable, but the FPS is at the limits of the usability and doesn't offer a good user experience.
Rhinoceros is instead a great surprise: despite it isn't its main role to manage textures and meshes, the navigation is not so bad, even if its frame rate isn't high and under the limits of a good usability.
Finally, it is important to underline that Rhinoceros imports every single texture sub-model as a separated object, that could be a problem during its use.
The FPS in Agisoft, tracked enabling only the model, is stable, but the FPS is very low, thus having a poor user experience. A deep analysis shows that Agisoft has a high processor utilization and only about 30% of Video Card utilization; this explains why it has so poor performance: it doesn't use principally the Video Card to manage the rendering, that it is much faster than the central CPU.
MeshLab is quite unusable with this model: the movement is step by step and each step require almost a second; MeshLab seems to suffer the same previous described issue of Agisoft.

fluidity; when the FPS drops down 30 FPS every people detects poor fluidity, stuttering, motion sickness and so on; under 15-20 FPS there isn't any more a video, but the user sees only a series of jerky images.

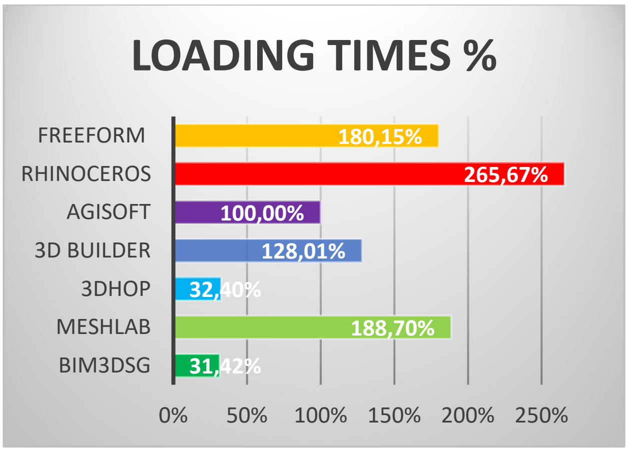


Figure 10: Loading times in percent, setting Agisoft as 100%; lower is better

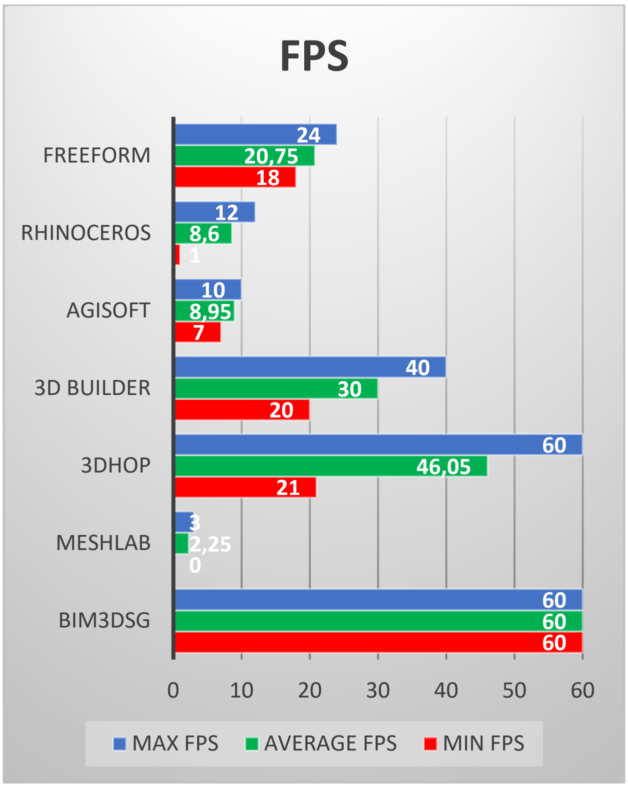


Figure 11: FPS of Umm al-Dabadib's Forte; monitor locked at 60 FPS, higher is better

Comparing only the average FPS and setting Agisoft as the reference (100%), because it is the software with which the model is created, the difference (Figure 12) between software are even more clear. BIM3DSG outperform Agisoft of 670,39% and it is better than Freeform by 289%. MeshLab has only 25,14% of the performance of Agisoft, that already doesn't do well, and BIM3DSG does 2.666% better than it.

5.4 Huge multi-texture & single texture models

This test is based on a part of San Marco Cathedral in Venice (Figure 13), loading about 700 textured objects of very different dimensions and textures resolutions, for a total texture size of 11.978.932.224 pixel (11,98Gpixel). BIM3DSG succeeds to load all the objects and correctly manages the scene; the navigation is good, with a framerate between 32 and 60 fps (monitor locked refresh), with an average of 45,85 FPS.

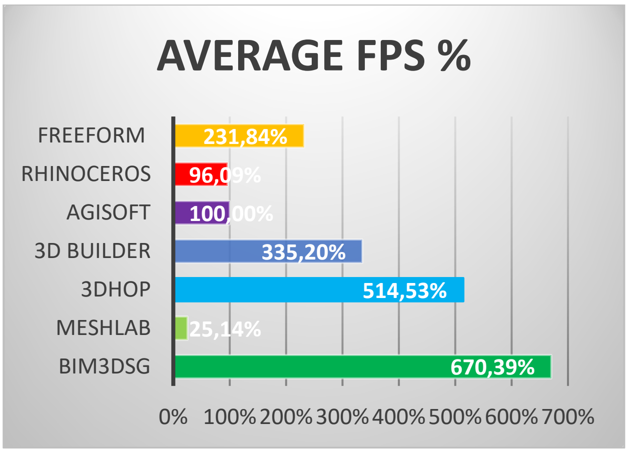


Figure 12: FPS in percent, settings Agisoft as 100%; higher is better

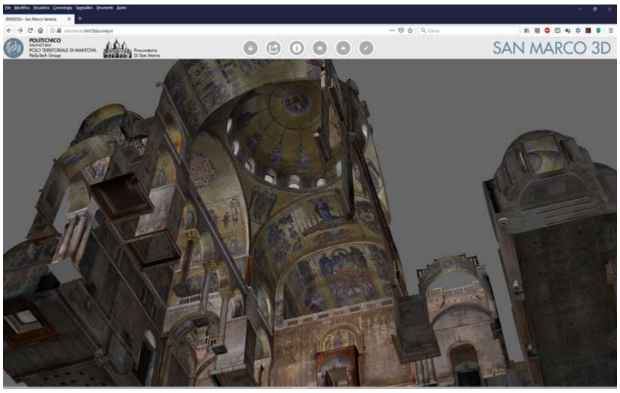


Figure 13: The used part for the test of San Marco in Venice

To load the scene, it is used 48GB of System RAM and quite all the Video RAM; the Video Card utilization also go over 95%, remarking the fact that BIM3DSG is able to use all the system available resources and using a faster GPU lets to have even better performance.
It is also possible to load the whole Cathedral, but it is need to half the best textures resolution size (use LoD1 for the textures) or to use a Video Card with 16GB Video RAM.

5.5 Scalability

5.5.1 Tablet test

This test is based on a grid of Saqqara's archaeological site, composed by two multi-textures models, with a total textures size of 150.994.944 pixel (150,99Mpixel).
The system succeeds to load all the objects and correctly manages the scene, even if it is working near the devices limits, using 1,2GB of RAM; the navigation is still acceptable, with a frame rate between 20 and 30 FPS.
If only the large model is loaded, the texture size became 83.886.080 pixel (83,87Mpixel) and the frame rate is stable over 30fps.

5.5.2 Smartphone

For this test it is always used the Forte of Umm al-Dabadib, but it is needed to load it with textures at LoD2, so 25 texture of reduced size of 1.024x1.024 pixel each, for a total of 26.214.400 pixel (26,21Mpixel), and model at LoD4 (0.4 of percentage reduction).
The result is impressive, considering the low available resources: the system loads all the object and correctly manage the scene, even if 1,4GB of RAM is used by the browser; the navigation with touch is quite smooth.

5.6 BIM3DSG OBJ Importer: import time

As previously described, the time depends principally on the MeshLab elaboration and eventually the upload speed; to reduce the impact of the internet connection, for this test it is used a good fiber connection, that lets to fast upload the 858MB of the OBJ, plus all 569 JSON files, needed for the 8 LoDs, and the 200 texture files.
To finish the entire, including the upload of all the needed files, process of importing the Forte of Umm al-Dabadib, the system takes 20 minutes and 13 seconds; it is important to underline that about 7 minutes are taken by MeshLabServer only to open seven time the OBJ, plus the time needed for applying the filter and save it again, that confirms that the most part of the time are due to the MeshLabServer’s elaborations.

6. CONCLUSIONS

The available technologies let to produce very high-quality reality-based 3D digital reconstruction, that are often required in CH field characterize by irregular and not-standard shapes. However, there may be problems to manage this type of model (and also to share with other professional figures) and it is fundamental to not waste precious resource. The use of multi-textures lets to have very high texture resolution for largest objects or when it is needing to represent very small details. However, the resources aren’t infinite, thus it is fundamental to precise calculate the textures size and to use the right resolution relate to the physical size of the real object and the survey resolution.
A comparison of most used software reveals how all of them fall in great difficulty when there is to manage huge models and textures: they aren’t able neither to give users a good experience and to rightly use all the available system resources; all of them suffers of motion sickness, low frame rate, stuttering and high loading times.
BIM3DSG of 3D Surver Group of Politecnico di Milano, a system specifically thought for the management, the sharing and the fruition of very high-resolution models, instead performs very well and it hasn’t any problem to deal with the tested models and scales very well also non low level devices; furthermore, it has a very flexible information system that lets to insert every information and data, plus some useful tool, such as measurement tools, orthophotos and hotspot supports. For these reasons, according to the presented test, it should be a very performant tool for management and web sharing of 3D digital reconstruction in CH field.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the LIFE (Living in a Fringe Environment) project of the professor C. Rossi and the 3D Survey Group for providing the used datasets.

REFERENCES

Adami, A., Fassi, F., Fregonese, L., and Piana, M., 2018. Image-based techniques for the survey of mosaics in the Basilica di San Marco in Venice. *Virtual Archaeology Review*, Vol. 9(19), pp. 1-20.

Bolognesi, C. M., and Fiorillo, F., 2018. Optimization of texture mapping process in the Reality-Based Modeling application. In: *Rappresentazione materiale/immateriale. Drawing as (in)tangible representation*, Gangemi, pp.337-342.

Di Benedetto, M., Ponchio, F., Malomo, L., Callieri, M., Dellepiane, M., Cignoni, P. and Scopigno, R., 2014. Web and Mobile Visualization for Cultural Heritage. In: *Lecture Notes on Computer Science* LNCS 8355, Springer, pp. 18-35.

FARO Technologies, 2018. SCENE WEBSHARE CLOUD, https://www.faro.com/products/construction-bim-cim/scene-webshare-cloud/ (21/12/2018).

Fassi, F., Achille, C., Mandelli, A., Rechichi, F. and Parri, S., 2015. A new idea of BIM system for visualization, web sharing and using huge complex 3D models for facility management. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XL-5/W4, pp. 359-366.

Fassi, F., and Campanella, C., 2017. From daguerreotypes to digital automatic photogrammetry. Applications and limits for the built heritage project. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-5/W1, pp. 313-319.

Fassi, F., Fregonese, L., Adami, A., and Rechichi, F., 2017. BIM system for the conservation and preservation of the mosaics of San Marco in Venice. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-2/W5, pp. 229-236.

Fassi, F., and Parri, S. 2012. Complex Architecture in 3D: from survey to web. *International Journal of Heritage in the digital era*, Vol. 1, pp 379-398.

Fassi, F., Rechichi, F. and Parri, S., 2014. Metodo e sistema per la gestione e la visualizzazione di modelli di oggetti tridimensionali complessi. In: Italian patent pending MI2014A002016.

Gonzalez-Aguilera, D., Remondino, F., Rodríguez-Gonzálvez, P., and Nocerino, E. (Eds.), 2018. *Remote Sensed Data and Processing Methodologies for 3D Virtual Reconstruction and Visualization of Complex Architectures*. MDPI.

Leica Geosystems, 2018. Leica TruView Family software, https://leica-geosystems.com/products/laser-scanners/software/leica-truview (21/12/2018).

Rechichi, F., Mandelli, A., Achille, C., and Fassi, F., 2016. Sharing high-resolution models and information on web: the web module of BIM3DSG system. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*., Vol. XLI-B5, pp. 703-710.

Rechichi, F., 2018, ITinnovations Library, online at https://www.itinnovations.it, (Last accessed on Dicember 2018).

Remondino, F., Nocerino, E., Toschi, I., and Menna, F., 2017. A critical review of automated photogrammetric processing of large datasets. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-2/W5, pp. 591-599.

SceneJS, online at http://scenejs.org (Last accessed on Dicember 2018).

Verhoeven, G. J., 2017. Computer Graphics meets image fusion: the power of texture baking to simultaneously visualise 3D surface features and colour. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. IV- 2/W2, pp. 295-302.

Visual Computing Laboratory - ISTI – CNR, 2018. 3DHOP (3D Heritage Online Presenter), http://vcg.isti.cnr.it/3dhop/ (21/12/2018).

Web Persistent Storage, 2018, online at http://www.w3.org/TR/webstorage (Last accessed on Dicember 2018).