VIRTUAL CLONES FOR CULTURAL HERITAGE APPLICATIONS

M. Potenziani, F. Banterle, M. Callieri, M. Dellepiane, F. Ponchio, R. Scopigno*

Abstract

Digital technologies are now mature for producing high quality digital replicas of Cultural Heritage (CH) artifacts. The research results produced in the last decade have shown an impressive evolution and consolidation of the technologies for acquiring high-quality digital 3D models, encompassing both geometry and color (or, better, surface reflectance properties). Some recent technologies for constructing 3D models enriched by a high-quality encoding of the color attribute will be presented. The focus of this paper is to show and discuss practical solutions, which could be deployed without requiring the installation of a specific or sophisticated acquisition lab setup.

In the second part of this paper, we focus on new solutions for the interactive visualization of complex models, adequate for modern communication channels such as the web and the mobile platforms. Together with the algorithms and approaches, we show also some practical examples where high-quality 3D models have been used in CH research, restoration and conservation.

Riassunto

L'enorme evoluzione e miglioramento delle tecnologie di digitalizzazione e modellazione 3D ci permette di creare in repliche digitali di alta qualità delle opere di interesse per gli operatori del settore del Patrimonio Culturale. Le ricerche sviluppate negli ultimi due decenni hanno fornito le basi per un notevole consolidamento delle tecnologie per la produzione di modelli in 3D, che oggi permettono non solo di derivare le caratteristiche geometriche ma anche il colore (o meglio, le proprietà di riflettività delle superfici). In questo articolo si offre una sintetica descrizione di alcune recenti tecnologie che permettono di codificare in modo sofisticato l'attributo colore e riflettenza su modelli 3D. Lo scopo di questo articolo è di presentare e discutere alcune soluzioni pratiche, sia di semplice uso a livello di setup di acquisizione che supportate da software open-source.

La seconda parte del lavoro ha come focus il problema della visualizzazione e disseminazione dei modelli 3D prodotti. Considerando sia il web che le piattaforme mobili come veicoli ideali per la disseminazione ed il largo uso delle risorse digitali prodotte, si presentano alcune nuove soluzioni per la visualizzazione interattiva di modelli complessi adatte a questo contesti di presentazione visuale. Insieme ad algoritmi e strategie, dimostreremo anche alcuni esempi dove i modelli in 3D sono usati nello studio, nel restauro e nella conservazione di beni culturali.

The progress of digital 3D acquisition and visual presentation technologies has been impressive. Scholars in Digital humanities (DH), curators and restorers dispose of excellent technologies for producing extremely accurate digital models of the objects of interest and for publishing and visualizing them in a wide and pervasive manner.

The goal of this paper is to review some enabling technologies developed by the Visual Computing Lab of CNR-ISTI; most of these solutions have been disclosed to the CH/DH community as open source resources and therefore are easily available for test and experimentation.

The domain of visual technologies for CH is very wide nowadays. To reduce the scope of this paper, we will review here only some recent approaches for: (a) the acquisition and mapping of color or surface reflectance data; and (b) the technologies and instruments for the interactive visualization of 3D models on the web and on mobile platforms.

The progress of digital 3D acquisition and visual presentation technologies has been impressive. Scholars in Digital humanities (DH), curators and restorers dispose of excellent technologies for producing extremely accurate digital models of the objects of interest and for publishing and visualizing them in a wide and pervasive manner.

The goal of this paper is to review some enabling technologies developed by the Visual Computing Lab of CNR-ISTI; most of these solutions have been disclosed to the CH/DH community

*Istituto di Scienza e Tecnologie dell'Informazione (ISTI), Consiglio Nazionale delle Ricerche (CNR) Pisa (Italy).

as open source resources and therefore are easily available for test and experimentation.

The domain of visual technologies for CH is very wide nowadays. To reduce the scope of this paper, we will review here only some recent approaches for: (a) the acquisition and mapping of color or surface reflectance data; and (b) the technologies and instruments for the interactive visualization of 3D models on the web and on mobile platforms.

Color Data Registration and Mapping

A precise, detailed geometry is essential for the study and documentation of a CH artifact, but this knowledge by itself does not represent a sufficient digital 3D model. Essential information also comes from the color or, better, the appearance data.

The reproduction of *apparent color*, obtained by mapping photographic information on a 3D model, is still the most common approach to describe the visual appearance of an object, especially in the cultural heritage field.

A robust solution is the direct use of images to transfer the color to 3D surfaces. In these cases, the apparent color value, as sampled in digital photos, is mapped on the digital object's surface by first registering these photos w.r.t. the 3D model (by estimating the camera parameters), and then applying an inverse projection and a selection or an integration rule, to compute surface color from the available pixel samples contained in the input images.

The first step in the color projection pipeline is *image registration*, since in most cases, the camera parameters associated with each image are not known in advance. Several automatic[1] and semi-automatic[2] methods for image- to-geometry registration have been proposed. They are mainly based on an analysis of the geometric features of the model (e.g., silhouette and orthogonality), or on some input given by the user (2D-to-3D correspondences).

The second phase is how to *select* or *blend* the information contained in the input images, with the aim of producing a good quality color-per-vertex or texture-based encoding of the color data on the 3D surface (this is usually know as the *mapping* stage).

Redundancy in input photographs is a resource but also a potential problem: we have to deal with the discontinuities caused by color differences between photos that cover adjacent areas, and to reduce illumination-related artifacts (i.e., shadows, highlights, and peculiar reflection effects).

To solve these problems, one group of methods selects, for each part of the surface, a portion of a representative image following a specific criterion - in most cases, the orthogonality between the surface and the view direction[3]. However, artifacts caused by the lack of consistency between overlapping images are visible on the borders between surface areas that receive color from different images. These can be partially removed by working on the border between two images[4].

An alternative solution ”blends” the contribution of all the images by assigning a weight to each one or to each input pixel (this value expresses the ”quality” of its contribution), and selecting the final surface color as the weighted average of the input data, as in[5]. The weight is usually a combination of various quality metrics[6]. In particular, a flexible and extendable weighting system was presented in[7].

We present in the following subsections how those phases (registration and merging) have been implemented in MeshLab.

MeshLab[8] is a multi-platform open source tool for the visualization and processing of 3D models. It is oriented to the management and processing of large, unstructured triangular meshes and point clouds, and it provides a set of tools for measuring, checking, cleaning, healing, inspecting, rendering and converting 3D meshes.

A. Image registration in MeshLab

In order to support the management of photographic information, a new data type has been introduced in MeshLab: the *raster layers*, which support loading of a series of images (Fig. 1).

These layers do contain a raster image, plus the camera parameters needed to establish a correspondence between the 3D geometry and the 2D image. These parameters describe the position and orientation (extrinsic parameters) and internals of the camera, like sensor size, lens distortion and focal length (intrinsic parameters) at the moment of the shot.

By obtaining these parameters it is possible to reconstruct the perspective projection that created the photo. This opens up two possibilities: being able to see the 3D scene trough the same camera that took the shot (thus, exploring the photographic dataset spatially), and project back the color information onto the 3D model (to generate color mapping).

The most robust approaches to image alignment are based on setting some correspondences between the 3D model and the ima-

[1] Ikeuchi, Oishi, Takamatsu, Sagawa, Nakazawa, Kurazume, Nishino, Kamakura, Okamoto 2007, pp. 189-208; Brunie, Lavallée, Szeliski 1992, pp. 670-675; Lensch, Heidrich, Seidel 2000, pp. 317-452; Wolberg, Zokai 2006, pp. 2293-2300; Corsini, Dellepiane, Ganovelli, Gherardi, Fusiello, Scopigno 2013, pp. 91-111.

[2] Franken, Dellepiane, Ganovelli, Cignoni, Montani, Scopigno 2005, pp. 619-628.

[3] Callieri, Cignoni, Scopigno 2002, pp. 419-426.

[4] Callieri, Cignoni, Scopigno 2002, pp. 419-426; Bannai, Agathos, Fisher 2004, pp. 558-565.

[5] Pulli, Abi-Rached, Duchamp, Shapiro, Stuetzle 1998, imaging 11-15.

[6] Bernardini, Martin, Rushmeier 2001, pp. 318-322.

[7] Callieri, Cignoni, Corsini, Scopigno 2008, pp. 464-473.

[8] Cignoni, Callieri, Corsini, Dellepiane, Ganovelli, Ranzuglia 2008, pp. 129-136.

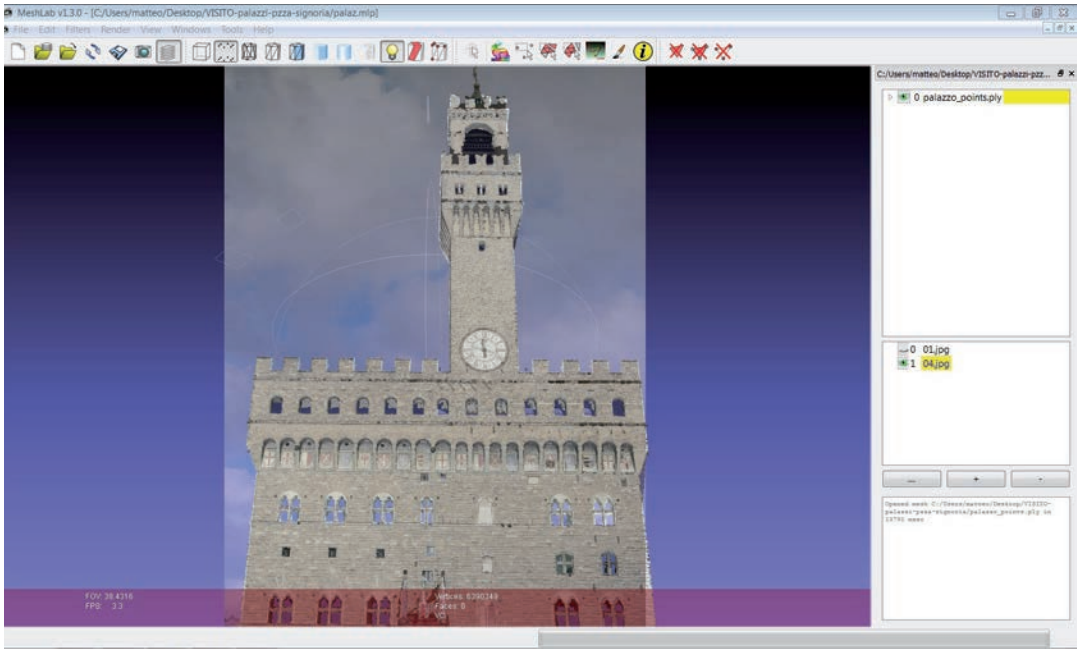

Fig. 1. Aligning an image to a 3D model.

ge[9]. Yet, this point picking operation may be tricky depending on the characteristics of the 3D object. Hence, a different approach, based on Mutual Information[10] was proposed. This is based on the calculation of a statistical measure of correlation between the image and a rendering of the model and has also been implemented in MeshLab.

Using an easy to use visualization, it is possible to have a direct feedback about the quality of the alignment. This alignment strategy works on triangulated 3D models but also on point clouds, making it quite versatile. An implementation of this method[11] is also supported in MeshLab (unfortunately it is not included in the official executable version, but it is included in the official code; users willing to use this version should either recompile MeshLab or contact us).

B. Color data mapping in MeshLab

Once the input photographs have been aligned, we need to map the color information onto the geometry. To this aim, MeshLab offers different color mapping tools to better cope with the different needs of the various datasets. By using the color data from the calibrated images, it is possible to generate detailed, artifact free per-vertex color encoding, fill an existing texture parametrization or generate an entirely new texture mapping, driven by the photographic coverage.

While the management of 3D geometry is more or less standard among the various processing software for sampled 3D data, the support of color information is still quite different from one tool to another. In addition, different file formats support different methods of color storage, adding complexity to the task.

The two main methods of color storage are *per-vertex* encoding (each vertex of the mesh has an associated RGB value) or *texture mapping* (where the color is mapped in a 2D image and we have a parametrization linking 2D and 3D spaces).

Per-vertex encoding works well for highly dense models, such as the ones produced by 3D scanning; this because the color detail is limited by the resolution of the geometry. It is a compact and effective way to store color, it is simple to use and works with most 3D modeling and rendering tools.

On the other hand, texture map is a more standard solution, decouples the resolution of the geometry from the resolution of the color detail and is supported (with small variations) by all 3D software.

While the texture approach seems more usable and standard in the field of 3D graphics, it requires a parametrization (a correspondence between a point on the 3D surface and a point on the 2D plane). This requirement is difficult to fulfill in the case of digital models coming from 3D scanning.

[9] FRANKEN, DELLEPIANE, GANOVELLI, CIGNONI, MONTANI, SCOPIGNO 2005, pp. 619-628.

[10] CORSINI, DELLEPIANE, PONCHIO, SCOPIGNO 2009, pp. 1755-1764.

[11] DELLEPIANE, SCOPIGNO 2013, pp. 39-46.

Fig. 2. The two color data encoding modes: per-vertex-colors (above) and texture mapping (below).

These models, because of their geometric complexity and their unstructured nature (they are, basically, just a bunch of unorganized triangles, often with topological problems), are quite difficult to parametrize, posing serious limitation to the texture map approach, especially for complex 3D models (more than 1 million triangles).

MeshLab offers both *color-per-vertex* and *texture mapping* modes (see Fig. 2).

Concerning the texture mapping approach, MeshLab can generate a quite trivial parametrization (each triangle packed on its own in the texture space), usable on simple objects, or an extremely advanced global iso-parametrization, with optimal properties in terms of low distortion

Concerning *the per-vertex* encoding, MeshLab offer a flexible *pixel blending* solution: all pixels in the input images are projected and blended over the 3D object surface. MeshLab offers an improved version of the weighted blending method described in[12], with a more efficient and fast computation, a smaller memory footprint and some tweaking in the weighting functions.

The method computes a color for any point over the surface, by estimating the most correct color as a weighted mean, which takes in account various quality metrics.

Each pixel of each input images has a quality value associated to it; this quality value is calculated using multiple metrics, like the distance of the camera to the sampled points (closer is better), the viewing angle (when the camera is more orthogonal to the surface, it is better) and the proximity of the pixel to critical points on the photo (border pixels are bad, and so are pixels close to depth discontinuities).

This meaningful and smooth weighting system ensures the weighted blending will produce a continuous, detailed and smooth color mapping. This color blending technique can be used both to generate per-vertex color (by evaluating the color blending for each vertex) or to fill a texture map associated to an existing parametrization (evaluating the color blending for each texel). This makes the method flexible, enabling the user to choose the most appropriate output format.

PRACTICAL REFLECTANCE ACQUISITION

A more complex representation of the material or surface reflection properties of the object would be much more effective in applications oriented to the CH domain. The most correct way to represent the material properties of an object is to describe them through a reflection function (i.e. BRDF), which attempts to model the observed scattering behavior of a class of real surfaces.

A detailed presentation of its theory and applications can be found in[13]. Unfortunately, state-of-the-art BRDF calculation approaches rely on controlled and complex illumination setups[14]: this limits their application in the context of complex scanning projects (artifacts span largely different sizes; we have to cope with on-the-field acquisitions, such as those performed in museums; we have time and resources constrains).

Thus, most of the approaches reported in literature are not very practical when the focus is the low-cost acquisition of valuable CH artworks.

An approach has been recently proposed to allow more practical acquisition of reflectance data[15]. The aim of this work is to define a method that simplifies the acquisition phase of the object surface appearance and allows to reconstruct an approximated Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF) of an object with complex geometry.

The method starts from video sequences acquired with fixed but general lighting conditions. The method is composed by three steps: the approximation of the environment map of the acquisition scene, using the same object as a probe; the estimation of the diffuse color of the object; the estimation of the specular components of the main materials of the object, by using a Phong model.

All the steps are based on statistical analysis of the color samples projected by the video sequences on the surface of the object. The final output is suitable to be used with a 3D model of the object to obtain accurate and photo-realistic renderings. Although the method presents some limitations, the trade-off between the easiness of acquisition and the obtained results makes it useful for practical applications.

PRESENTING COMPLEX 3D MODELS ON THE WEB

Many approaches for supporting *interactive visualization* of complex models have been presented in the last ten years; our goal in this work is to focus on the technologies that allow implementing interactive manipulation and visualization of 3D models on the web and on mobile platforms.

Models produced with 3D scanning or photogrammetric solutions are characterized by a very dense sampling and thus very complex triangle meshes (or point clouds). Any interactive solution should therefore apply modern geometric simplification and multiresolution encoding technologies to ensure interactive throughputs without compromise on data quality.

This is even more urgent when we plan to use the web as our main distribution and sharing channel. Enabling technologies are therefore required to support easy access to multimedia representations on the web and high-quality visualization directly inside standard web pages.

The appearance of the WebGL standard in 2009[16] was a fundamental change. WebGL provides therefore a specification on how to render 3D data, that web browsers should implement. Hence, by incorporating the WebGL approach, modern web browsers are able to natively access the 3D graphics hardware without needing additional plug-ins or extensions.

The easy visualization of 3D models is an immediate result that can be produced with browsers that endorse WebGL.

Some examples in the CH domain are the access to a repository of 3D models produced by Fraunhofer IGD in the framework of the 3DCOFORM project (see at http://www.3d-coform.eu/x3dom/

[12] CALLIERI, CIGNONI, CORSINI, SCOPIGNO 2008, pp. 464-473.

[13] DORSEY, RUSHMEIER, SILLION 2007.

[14] LENSCH, KAUTZ, GOESELE, HEIDRICH, SEIDEL 2003, pp. 1–27; DEBEVEC, HAWKINS, TCHOU, DUIKER, SAROKIN, SAGAR, 2000, pp. 145-156.

[15] PALMA, CALLIERI, DELLEPIANE, SCOPIGNO 2012, pp. 1491-1500.

[16] Khronos Group. WebGL - OpenGL ES 2.0 for the Web (2009).

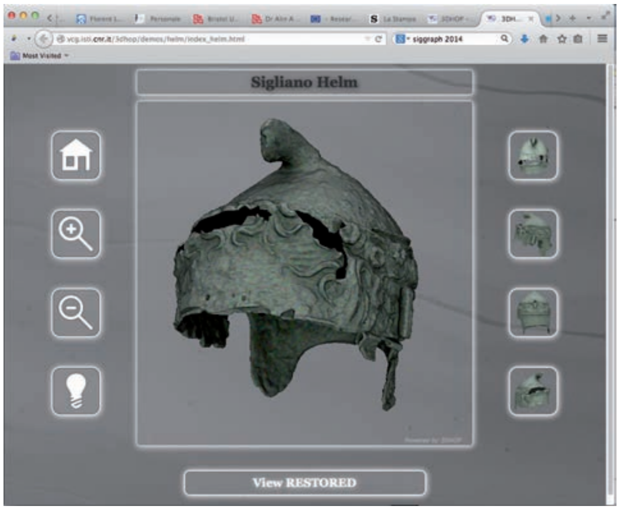

Fig. 3. One of the visual presentation layouts provided by 3DHOP.

index.html; consulted URL on 28.6.2020) or the re-design of the Cenobium system (see at http://cenobium.isti.cnr.it/; consulted URL on 28.6.2020).

Unfortunately, the design of web-based presentation which includes 3D content is possible with WebGL, but it is still not easy and usually very far from the technical capabilities of most CH users. To cope with this issue, we have recently designed and implemented a software framework, called 3DHOP (3D Heritage Online Presenter)[17].

The use of 3DHOP simplifies the creation of interactive visualization webpages, able to display high-resolution 3D models, with intuitive user interaction/manipulation; moreover, these resources can be deeply connected with the rest of the webpage elements (Fig. 3).

- The most interesting characteristics of the 3DHOP framework are:
- The ability to work with extremely complex 3D meshes or point clouds (tens of million triangles/vertices), using a streaming-friendly multiresolution scheme.
- The easiness of use for developers, especially those with background in web programming, thanks to the use of declarative-style scene creation and exposed JavaScript functions used to control the interaction.

The availability of a number of basic building blocks for creating interactive visualizations, each one configurable, but at the same time providing sensible defaults and comprehensive documentation.

3DHOP has been released as open source (GPL licence) in April 2014, and it is available to be tested and used. The downloadable package, with documentation, a series of tutorials (How-Tos) and a Gallery of examples is available at the website: http://3dhop.net (consulted 6 on 28.6.2020).

3DHOP has been designed with the aim of being easy to use, especially for people having a background in Web development, thus without requiring solid knowledge in CG programming.

3DHOP is not a "silver bullet", able to support any possible application or visual communication project, but a framework designed to deal with specific needs. It is an ideal tool to visualize high-resolution single objects (especially with dense models coming from 3D scanning (Fig. 3) or, more in general, a simple static scene composed by complex models (Fig. 4). Conversely, 3DHOP is not suited to manage complex scenes made of low-poly objects (this is a common case when working with CAD, procedural or hand-modeled geometries).

3DHOP makes possible a fast deployment process when dealing with simple interaction mechanisms, making it a good choice for quickly creating interactive visualization for a large collection of models. Additionally, 3DHOP integrates extremely well with the rest of the webpage, thanks to its exposed JavaScript functions. The ideal situation is having the logic of the visualization scheme in the page scripts, and use 3DHOP for the 3D visualization.

3DHOP tool has been designed with different levels of entry, to be as straightforward as possible for the more simple cases but, at the same time, able to provide enough configurable features to support the huge variability of Cultural Heritage artworks and applications.

Developers with limited programming skills may use the framework using one of the following strategies:

- *Zero configuration*: since all the components have a set of safe defaults, it is possible to create a visualization page without configuring anything. This "minimal" visualization page is contained in a folder of the distribution, and can be readily used by the most inexperienced of users, since it is only necessary to change the 3D model file.
- *How-Tos*: in addition to a plain documentation, we opted to present the different features with *How-To* descriptions, detailing the parameter-based configuration of the visualization component. These pages contain reusable

[17] Potenziani, Callieri, Ponchio, Scopigno 2015, pp. 129-141; Potenziani, Callieri, Dellepiane, Scopigno 2018, pp 244-333; Potenziani, Callieri, Scopigno 2018, pp. 169-178.

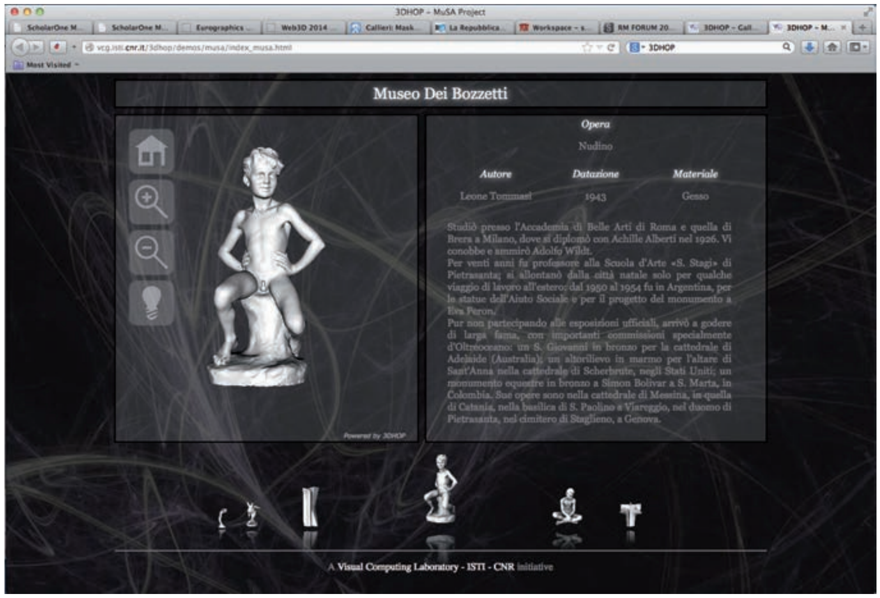

Fig. 4. The 3DHOP presentation layout designed for the interactive presentation of a small collection of artworks.

examples that can be modified following the content of the How-To resources. New resources will be added as soon as new features and components will be introduced in 3DHOP.
- *Templates*: in the Gallery page of the 3DHOP website, it is possible to find various examples (with different levels of complexity) which cover typical cases of usage in the CH field. The idea is to provide the developers with non-trivial usable templates, which can be used or customized with just minimal changes. After changing just the 3D model file (and the graphic elements, if needed), a completely unskilled developer may create its own visualization page without even modifying the HTML code. We are now working on a better documentation for the templates, and on the cleaning-up of their HTML code for a simpler use.

Moreover, many CH projects require ad-hoc solutions to cope with the specific requirements, to fully exploit the data and to reach the communication goals. In these cases, 3DHOP may be seen as a "codebase". The modular structure of the tool helps the implementation of new, specialized components reusing the existing 3DHOP modules, or their tuning.

Visualization on the Mobile Platform

Transmission and rendering of 3D models is possible also on the mobile platforms (smartphones and tablets), but the specificity of both the delivery channel and the presentation platform opens some issues.

Some of those issues are common to the web-based domain: efficient transmission of complex data should be ensured (mobile systems have stronger limitation on bandwidth than standard internet connections); efficient and interactive rendering should be supported, even on complex models (and thus multiresolution is required to sustain interactive rendering rates on complex models). Another issue specifically raised by the mobile platform is the need for efficient and easy-to-use manipulation interfaces.

We report here some experience gained in two recent projects.

CNR-ISTI introduced in 2012 a visualizer for 3D meshes running on smartphones or tablets, porting to this platform just the visualization component of MeshLab. This tool was released for both Apple iOS (http://www.meshpad.org/) and Android (https://play.google.com/store/apps/details?id=it.isticnr.meshlab).

The goal was to implement a mobile viewer for 3D models (single objects, rather than complex architectures). The viewer reads a variety of file formats and supports interactive rendering of meshes of size up to around 2M faces. The manipulation interface follows the usual touch-based approach common on mobile apps. A set of interaction gestures was designed and implemented (see Fig. 5):
- a *one-finger-drag* gesture allows to orbit around the object, thus controlling the three DOF of rotation;
- a *two-finger-drag* gesture is adopted to drive X- and Y-axis translations;
- a *two-fingers shrink/pull-apart* gestures perform uniform scaling;

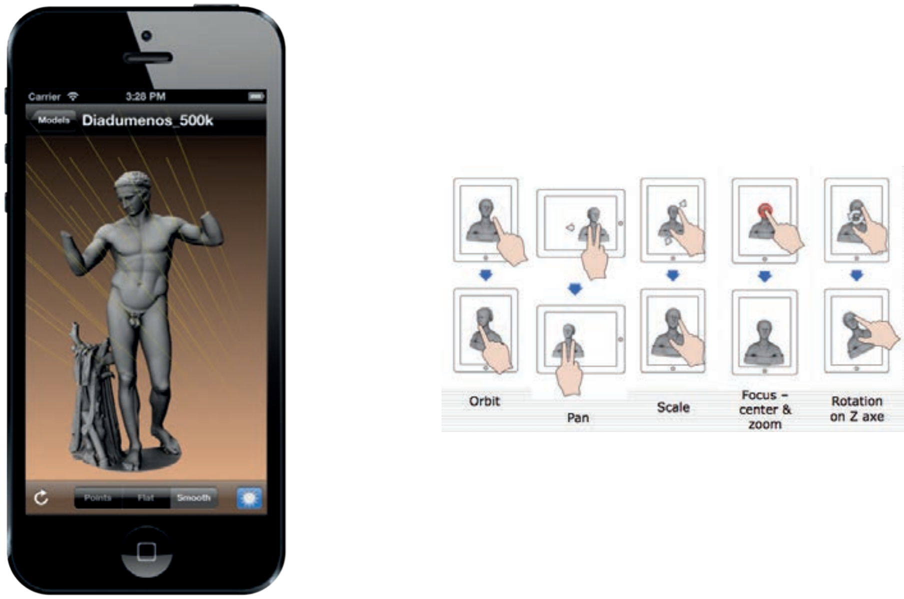

Fig. 5. The interface of MeshLab on iOS (left image) and the touch-based interaction dialogue (right image).

- a *two-finger rotate* gesture achieves local Z-axis rotation;

and, finally, a *single-finger double-tap* performs a change-of-focus operation, bringing the tapped point of the 3Dnsurface in the center of the screen and zooming on it.

These gestures come across as 'natural' because each finger interacting with the touchscreen correlates directly with the underlying 3D model visualized on the display or, in the case of the orbit operation, with a spherical proxy object that acts as a handle for the rotation of the model.

Manipulation of a single artwork is therefore easy, while free navigation in a reconstructed building or space is much more complicated in terms of design of the GIU, especially when we do not have the possibility to use sophisticated input devices.

This is again the case of the mobile platform, where we could also be interested to port a virtual navigation application. One specific application we designed recently is *VirtualTour*, an app for the iOS platform, supporting the easy and natural exploration of cultural heritage sites captured with 3D scanning technologies or modeled by artists.

The focus of this app is to provide a mobile access to hidden or inaccessible sites (caves, temples, buildings, etc), using 3D representation and breaking the usability barrier that often hinders the navigation in complex models.

In *VirtualTour* we have to drive the navigation in a 3D space under some constraints: (a) we do not have any input device but the touch screen and (b) we cannot us sophisticated I/O devices. The solution provided in *VirtualTour* is based first on the definition of a pre-defined path (connecting the entrance of the building and to its main rooms).

The choice of a constrained navigation, while somehow limiting the possible viewpoints, greatly simplifies the interaction with the reconstructed architecture, providing an interaction that is tailored to its shape.

Therefore, navigation is over a constrained path, with the selection of the location over this path guided by a simple slider and a default navigation speed, while the selection of the view direction driven by the device sensors. *VirtualTour* uses the mobile device sensors, such as the gyroscope and accelerometer, to provide intuitive and natural interaction.

For example, the user can rotate the camera view, while exploring the virtual site, by rotating the device and exploiting the device's gyroscope to gather the specific rotation and use this data for the rendering. In this way, the tablet becomes a window into the virtual word, with a behavior very easy to grasp even for naïve users.

Moreover, the user can either follow the predefined path at constant speed, or can drive the movement in the virtual space by walking in the real-world (his/her steps are detected, using the accelerometer, and again this data is used to feed the rendering engine, allowing to walk around in the virtual site). 3D models are rendered by using a state-of-the-art real-time rendering library developed by CNR. The current version of our rendering engine (with

Fig. 6. The interface of the Virtual Tour (an iOS app) designed to navigate a restricted access archaeological monument.

a plain model, not using multiresolution) allows to render models up to 2Mtr on commodity mobile platforms.

This paper has reviewed two main topics related with the production and visualization of 3D models for CH applications. We have described here the functionalities of the MeshLab platform for the management of the color information in conjunction with high-resolution 3D models (image registration and color mapping).

We have also briefly touched the issue concerning web presentation of complex 3D models and presented 3DHOP, a framework that aims at providing an easy way to create advanced 3D web content, offering the possibility to create and share advanced examples. Its modular structure has been designed to allow different utilization levels of the framework but also to enable the creation of a community of users, so that examples and new components may be shared and re-used.

We believe that this could be a helpful instrument to help the CH community to create and share advanced contents on the web, and use it not only for dissemination purposes, but also in the workflow of experts and practitioners. Finally, we have presented a couple of examples of 3D visualization over the mobile platform.

All those technologies give us impressing and unprecedented capabilities for documenting CH artworks and for telling their story. Their potential impact on the way we perceive or we are educated on artistic themes is impressive. Those technologies could bring a revolution in the way people document, communicate or teach arts[18].

[18] The research presented here has been funded by European project "ARIADNE Plus" (FP7–INFRA–2018–1–823914; http://www.ariadne-infrastructure.eu/ (consulted URL on 28.6.2020).