Web and Mobile Visualization for Cultural Heritage

Marco di Benedetto, Federico Ponchio, Luigi Malomo, Marco Callieri, Matteo Dellepiane, Paolo Cignoni, and Roberto Scopigno

Visual Computing Lab, CNR-ISTI, via Moruzzi 1, 56124 Pisa, Italy
{marco.dibenedetto,federico.ponchio,luigi.malomo,
marco.callieri,matteo.dellepiane,paolo.cignoni,
roberto.scopigno}@isti.cnr.it

Abstract. Thanks to the impressive research results produced in the last decade, digital technologies are now mature for producing high-quality digital replicas of Cultural Heritage (CH) artifacts. At the same time, CH practitioners and scholars have also access to a number of technologies that allow distributing and presenting those models to everybody and everywhere by means of a number of communication platforms. The goal of this chapter is to present some recent technologies for supporting the visualization of complex models, by focusing on the requirements of interactive manipulation and visualization of 3D models on the web and on mobile platforms. The section will present some recent experiences where high-quality 3D models have been used in CH research, restoration and conservation. Some open issues in this domain will also be presented and discussed.

Keywords: Web-based graphics, digital 3D models, mobile platforms, interactive visualization and navigation.

1 Introduction

The progress in optical systems and visual computing has produced a number of mature technologies for producing high-quality digital 3D replicas of Cultural Heritage (CH) artifacts. Given the progressive availability of accurate and information dense digital 3D models, the web and the mobile devices are ideal platforms for the dissemination of those cultural assets.

Given the availability of technologies able to produce *high-resolution models* (with high-resolution we mean sampled models counting from 5M up to hundreds of millions faces/points) that are also characterized by high-quality mapping of the color or reflection properties [1, 2, 3], the issues are now: how to encode efficiently those data; how to archive and make them accessible to the community; and how to visualize them efficiently in the framework of CH applications [4].

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. Many view-dependent rendering

M. Ioannides (Ed.): 3D Research Challenges in Cultural Heritage, LNCS 8355, pp. 18–35, 2014.

solutions have been presented, able to process a multi-resolution model and extracting frame-by-frame view-dependent representations that fulfill the rendering quality and the performance constraints. Some of these solutions are now available on the entire spectrum of platforms (from desktop computers to tablets and smartphones, linked to the internet by wire or by mobile wireless connection), demonstrating that we dispose of a common enabling technology able to carry high-quality graphics to everyone and everywhere. Section 2 will focus on this theme.

Human-computer interaction is another major issue, specifically when the focus is a domain where users are usually not ICT professionals. If we focus on CH applications, then the potential users are museum visitors, CH curators, restorers, scholars; therefore, we cannot assume that the average skill in manipulating digital 3D objects or scenes will be the one we experience in the gaming domain or in any other domain where fluency with Computer Graphics interaction technique is a prerequisite skill. Interaction with 3D objects or scenes usually is not a simple task, due to the not uniform management approaches used by different systems and the risk of losing yourself in the void space while manipulating or navigating a virtual scene. Section 3 presents a new approach to the virtual manipulation of an object and for the interactive selection of view. We illustrate the results of an implementation of a navigation paradigm over a mobile platform that supports multi-touch interaction (incorporated in the "MeshLab for iOS" application, http://www.meshpad.org/). A number of preliminary evaluations, carried out with test subjects, and the feedback received from real users from different CH fields demonstrated that the multi-touch approach is much easier to use than the traditional mouse-based trackball. The result has been an impressively short time required to learn how to drive the interactive navigation around an object, including zooming and panning.

Integration of different media described in in Section 4 is the last issue that will be discussed. Virtual 3D models are just one of the available media that are used to document the status and the beauty of our CH assets. Images (both the standard ones and the more advanced 2D media such as RTI or panoramic images) play an important role; video is a resource easier and easier to acquire and to distribute to users. The improved insight that can be gathered by the use of multiple media should be taken into account, preventing CG people from stressing only the use of the 3D media. This means designing and developing technologies able to link different media or to present them in a coordinated or integrated manner. Several interesting approaches have been presented recently: web systems which allow to present and inspect 2D and 3D representations (e.g. the Cenobium system, http://cenobium.isti.cnr.it/ [5]; visualization tools that allow to inspect and analyze different types of images and 3D data (e.g. the YALE open source visualization system [6]); and systems able to navigate interactively a 3D scene and a set of geo-located 2D images (e.g. the PhotoCloud system, http://vcg.isti.cnr.it/photocloud/). Finally, another major advance would be also to create a more tight and coherent relation between any *digital model* and the *text* that encodes our knowledge on the story and meaning of the represented artwork. We will present some preliminary results on the design of a system able to support links from the text to the 3D object

and vice-versa, providing therefore an unprecedented tight integration between these two media.

2 Interactive Rendering of Complex Models

The quality of current 3D sampling methodologies makes the reconstruction of high-resolution digital models of the artworks of interest an off-the-shelf capability. Standard short-range acquisition devices allow to produce at least 4-10 samples for squared millimeter, thus leading to digital models composed by several millions or tens of millions sampled points. Long range technologies support coarser sampling densities per squared unit, but since the sampled surface extension is usually much larger, models with up to hundreds of million samples are also common. Even Multi-view stereo techniques are now able to handle thousands of images, and provide extremely dense 3D models.

The value of the models reconstructed with modern sampling technologies resides in their accuracy and density, meaning that we have a huge quantity of geometric data to be used in applications. But, at the same time, all those data can be a problem for implementing efficient computations or visualization; a very common concern raised very often in the near past was that the density of sampled models made it impossible to use them in real applications. Therefore, endorsing methods able to produce a controlled granularity reduction of the digital models is mandatory in CH applications.

2.1 Simplification and Multiresolution Management of Complex Models

Many years of research in computer graphics have been instrumental in building an arsenal of technologies for controlled surface simplification and for the construction of multiresolution encoding schemes and view-dependent rendering modalities [7, 8]. The work at CNR-ISTI led to the design of several approaches and tools that contributed to this evolution. Most of them have been distributed to the community. Efficient tools for the simplification of triangulated surfaces are nowadays supported in MeshLab (http://meshlab.sourceforge.net/) and are also at the base of a multiresolution representation and rendering library, called ***Nexus*** (http://vcg.isti.cnr.it/nexus/).

Nexus is a multiresolution visualization library supporting interactive rendering of very large surface models. It belongs to the family of cluster based, view-dependent visualization algorithms (see for example the Adaptive Tetrapuzzles [9] and the Batched Multi Triangulation (BMT) [10] approaches). It employs a *patch-based approach*: the granularity of the primitive elements is moved from triangles to small contiguous portions of a mesh (patches composed by a few thousand triangles), to reduce the number of per-element CPU operations. Moreover, a batched structure allows for aggressive GPU optimization of the triangle patches; the latter are usually encoded with triangle strips, boosting GPU rendering performances.

To hold and manage the large number of alternative patches that compose the multiresolution encoding, Nexus adopts a spatial partitioning strategy based on KD-trees, which combines fast streaming construction of the multiresolution model with efficient adaptive spatial partitioning of the mesh.

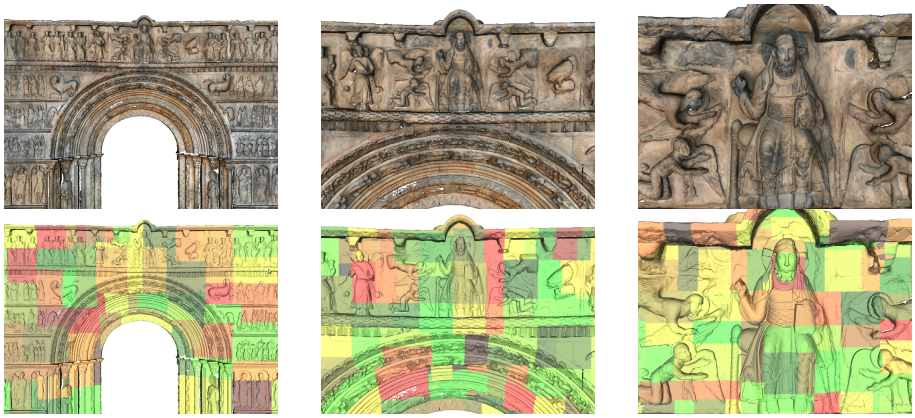

Fig. 1. An example of view dependent rendering by means of the Nexus library: the images on the top show different frames of a zooming-in navigation (moving towards a selected region of the surface) over the 3D model of the Portalada of Ripoll (Spain); the images below show the subdivision in patches of the view-dependent representations used to produce the images above

The Nexus multiresolution encoding is built from a triangle soup (or a point-based representation), by following iteratively the steps below:

1. The triangles from the triangle soup stream are inserted into a space subdivision structure (based on a KD-tree);
2. For each leaf of the KD-tree the triangles are collected, and a small mesh is generated and saved as a node in the multiresolution model;
3. The borders of the mesh (the triangles whose vertices do not fully belong to the node) are marked as read-only and the mesh is simplified using a high quality quadric-error simplification algorithm, leaving the patch borders unchanged;
4. Finally the triangles of the simplified mesh are pushed (and shuffled) into a new triangle soup stream, and the procedure is applied from the beginning again.

After each iteration, the number of triangles is halved and a new set of potential derivation between corresponding set of patches is constructed. These possible rules for replacement of patches are stored in a directed acyclic graph (DAG) and used at rendering time to produce the view-dependent representation.

Rendering a Nexus model requires a traversal of this DAG, to select an appropriate cut and the corresponding set of patches that will produce the view-dependent representation extracted from the current frame (see Figure 1). For each node we estimate the screen space error using a bounding sphere and the simplification error

recorded during the construction stage. Only the DAG has to be kept in RAM during the selection of each view-dependent representation. This means that a small memory size is needed to store the multiresolution structure. The data corresponding to the selected patches are loaded from disk on demand. This results in a very low number of OpenGL function calls (on the order of one thousand) even for large models and this allows to obtain efficient rendering performances.

The Nexus scheme supports the management of models with attached color data. The current Nexus version supports meshes adopting the *color-per-vertex* approach (this is common for very dense meshes, where the color per vertex encoding is more efficient than texture mapping). We are extending Nexus to support also triangle meshes having the color channel mapped by textures.

2.2 Complex Models on the Web

The web is now perceived as the main channel for accessing information through a wide variety of multimedia representations, and for managing the documentation of a CH artwork. 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 delivery of 3D content through the web comes with a considerable delay with respect to other digital media such as text, still images, videos and sound. Just like it already happened for commodity platforms, 3D content visualization is the latest of the functionalities acquired by web browsers.

Originally, 3D content was used only locally; nevertheless, remote visualization was perceived as an important feature, and thus collaborative and remote rendering solutions were proposed since the ‘90s. Then, several different approaches have been proposed for distributing and visualizing 3D data on the web (e.g. VRLM, X3D); the disadvantage of those approaches was that they confined 3D data to a specific visualization tool, implemented as a plugin (i.e., binary executable modules external to the hosting browser) that had to be explicitly installed by users. This approach was not ideal for the CH community, where potential users are usually not ICT experts and where the appearance of a blank screen corresponding to a request of installation of a piece of software frequently discourages the user from further exploration.

Fortunately, the evolution of the web is helping us. The appearance of the WebGL standard in 2009 [11] was a fundamental change. WebGL is the newborn component of the OpenGL ecosystem, and it is modeled as a JavaScript Application Programming Interface (API) that exposes a one-to-one mapping to the OpenGL ES 2.0 specifications. 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. Since WebGL is a low-level API, a series of higher-level libraries have been developed to help both expert and non-expert users in the design and implementation of applications. They differ from each other by the programming paradigm they use, ranging from scene-graph-based interfaces, such as X3Dom [12], to procedural paradigms, like SpiderGL [13].

WebGL has been already used to implement complex rendering systems and different interaction modalities in several fields.

There is an enormous potential brought by WebGL for CH applications development. This is not just because we do not need anymore to install specific plugins, but also because WebGL allows 3D data to become one of the media that can be shown in a web page natively. We are no more confining digital 3D assets to the ghetto of the specific plugin, but we are immersing them in the full multimedia context.

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/index.html) or the recent re-design of the Cenobium system (see at http://cenobium.isti.cnr.it/ or Figure 2).

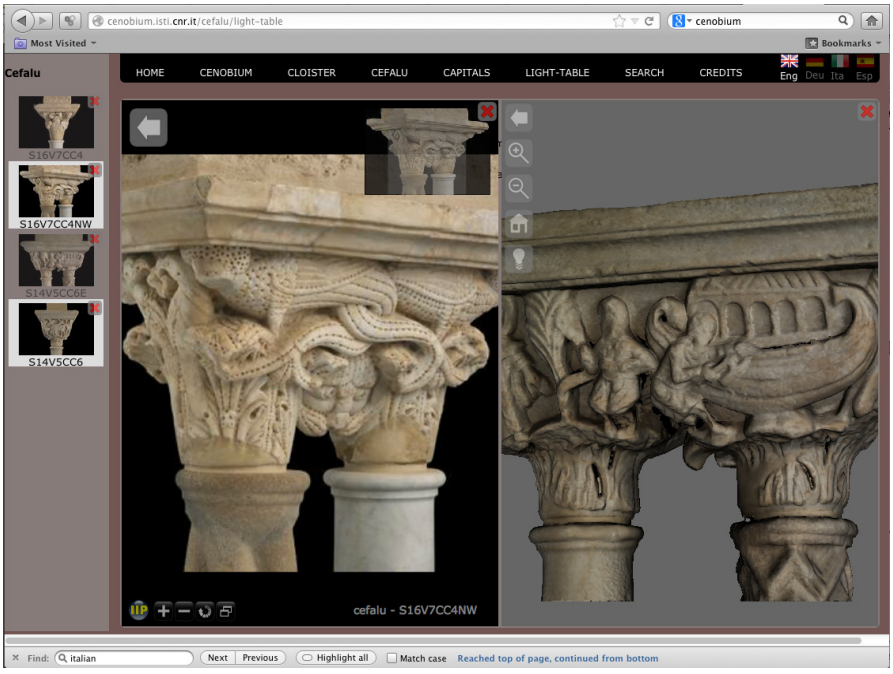

Fig. 2. An example of visual presentation provided by the Cenobium system [5]: in the same window we have a side by side visualization of a high-resolution image (left), and of a 3D model (right) of two different capitols from the Cefalù cloister (Sicily, Italy)

CH applications are very often based on complex 3D models: therefore, efficient transmission and visualization of 3D models becomes immediately an issue.

To efficiently render large 3D models on the web, we ported our Nexus approach to WebGL. The view-dependent rendering engine of the Nexus system has been ported from C++ to JavaScript on top of SpiderGL, revealing that JavaScript performances were not a limit due to the minimal processing required by the Nexus library (a simple traversal of the multiresolution hierarchy); the limitations of WebGL with respect to the more complex desktop OpenGL API were also not a problem due to the very basic OpenGL features required by Nexus. Obviously, using a multiresolution encoding introduces some penalty in terms of storage, since the multiresolution encoding is usually larger in size than the input mesh. We report in Table 1 some figures on the triangle meshes presented in this paper. To support fast data transfer we adopted a streaming approach over HTTP; Nexus objects are stored on a standard hard disk on our web server and served directly by any HTTP server (we used the Apache server). Being based on a multiresolution approach, the Nexus rendering engine sends data to the remote client on demand, i.e. low resolution data is transferred at the beginning of the visualization session and only the required details are transferred later on, following the user navigation and inspection needs. To further speedup the data transfer we have also enhanced Nexus with a data compression feature; the 3D data encoding tool is now able to perform geometric and topologic compression over the Nexus triangle patches. Unfortunately, this technology cannot be used in the Nexus porting for the WebGL platform, since the Javascript implementation of the client-side decompression stage is way too slow and degrades the performances, rather than improving them.

Table 1. Some figures on a few dataset (most of them shown in the paper figures): size of the input mesh and size of the Nexus encoding (without compression). Monreale capitol is one sample capitol presented in the Cenobium system, Monreale Cloister (http://cenobium.isti.cnr.it/). Michelangelo David is the model produced by the Digital Michelangelo project (https://graphics.stanford.edu/projects/mich/). Ruthwell Cross is the digital model of an 8th cent. Anglo-Saxon cross in UK. Finally, the Ripoll Portalada is the model of a large and carved church portal in Ripoll, Catalona, Spain.

	Plain input mesh		**Nexus encoding**	
Model	No. faces	Space (MB)	No. faces	Space (MB)
Monreale capitol	4.8 M	96 MB	9.5 M	159 MB
Michelangelo David	56 M	1.2 GB	112 M	1.9 GB
Ruthwell Cross	112 M	2.5 GB	224 M	3.7 GB
Ripoll Portalada	178 M	3.6 GB	354 M	5.8 GB

Nexus is the technology used to support most of our current projects deploying 3D on the web (e.g. the Cenobium system [5]). A recent result of the EC "3DCOFORM" project (http://www.3d-coform.eu/) was the Community Presenter (http://vcg.isti.cnr.it/presenter/). Community Presenter is a collection of tools and templates for the creation of multimedia interactive presentations of cultural artifacts, represented by means of their digital 3D model. The target audience is CH personnel (an art historian or a museum staff member with limited ICT experience or just assisted by an ICT professional). The Community Presenter allows easy visualization in HTML pages or QML applications of media such as 3D models, images, Reflection Transformation Images (RTI), video and audio. As a main feature, it supports the streaming of multiresolution 3D meshes over HTTP (using the previously discussed Nexus format), allowing the exploration of very large models. The code is based on declarative programming and extensive use of templates: this allows basic users to build presentation by simply filling in a few variable values (names of models, settings etc.) and without preventing advanced users from modifying interfaces and adding advanced functionalities. Community Presenter has already been used to design some informative systems, which were part of the assessment and dissemination actions of the 3DCOFORM projects. Some of these have been shown in the temporary exhibition that took place in Brighton on July-August 2012 (http://www.3d-coform.eu/index.php/dissemination/exhibitions).

One of these preliminary results is a multimedia kiosk that tells the story of a set of coins from the collection of the San Matteo National Museum (Pisa, Italy). Coins are an ideal test bed to show the potential of multimedia systems, since they are very small artworks and in a standard museum exposition they are presented to the public from a distance (typically at least 50cm from the observer's eye). This distance does not allow visitors to note the small and interesting details on the legend or on the carved decoration of the coins; moreover, coins are usually visible only from one side. Furthermore, coins have a lot of hidden knowledge that is difficult to transfer to visitors in an easy, effective and understandable manner. Therefore, the Community Presenter tool has been used to present those ancient coins in an innovative way [14], to better capture the interest of visitors and to give them enhanced information (see Figure 3). In the case of this multimedia presentation, we decided to use an advanced 2D image-based representation rather than a standard 3D medium. The requirement was to support the easy manipulation of the coins and the dynamic change of illumination, in order to allow the users to inspect those small artifacts in detail. This means that we needed a virtual representation capable of simulating illumination effects in real-time and in an accurate way, producing photo-realistic renderings. Therefore, to support easy and high-quality relighting of the digital coins, we decided to digitize them by acquiring RTI. RTI encoding is a computational photography method that, starting from a set of images taken from a single view under varying lighting conditions, encodes the object surface reflection (apparent color) by means of a function of the incident light direction. This encoding enables the interactive relighting of the object from any direction. A few snapshots from the coin multimedia kiosk are presented in Figure 3.

The results obtained with this installation let us broaden the practical use of in-browser 3D hardware acceleration: apart from efficiently using WebGL for 3D scenery visualization, we were able to exploit the power of the underlying graphics system to provide useful and complex effects even on 2D data (e.g. images), enabling a performance-oriented and cooperative interaction between Computer Vision and Computer Graphics methodologies.

Obviously, being able to include 3D data in web pages and supporting interactive rendering is not enough. Our skill and focus is on 3D interactive visualization, hence the content and focus of this section. But we would like to state explicitly here that the availability of 3D data on the web is a poor and incomplete result if a correct and complete management of the associated metadata is not supported and made accessible. The 3D model should therefore be paired by data that clarify *what* is the CH artwork represented and *how* the digital model was created (qualification of the digital model creation process). The absence or presence of these data makes the distinctions between a nice model and a trustable/usable digital document. We direct the interested readers to the many papers appeared in the last few years on the issues on metadata creation, management and searching [23,24] in the context of 3D data for the cultural heritage domain.

Fig. 3. Four snapshots taken from the multimedia kiosk of the San Matteo Museum (Pisa, Italy), showing different visualizations features provided by the coin viewer, like links to more detailed information (top right), navigation on high resolution images (bottom left) and interactive relighting (bottom right)

2.3 Porting 3D to the Mobile Platforms

Given the impressive technological evolution of the mobile platforms (smartphones and tablets) and their enormous commercial success, many applications are migrating towards the mobile domain. Transmission and rendering of 3D models is possible also on these platforms, 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 (see the next section for a brief discussion of this issue).

Smartphones or tablets could become very effective devices to present visual data. In the specific CH domain, this is definitely the case of the systems supporting the visitors of museums or of historical cities (several interactive virtual guides have been developed and experimented with recently based on portable devices). Another application could be assisting restorers in the annotation of the status of an artwork, or in the documentation of a restoration work. These portable devices could be the ideal platform for supporting the inspection and the annotation process (on top of digital representations that can be either 2D images or 3D models).

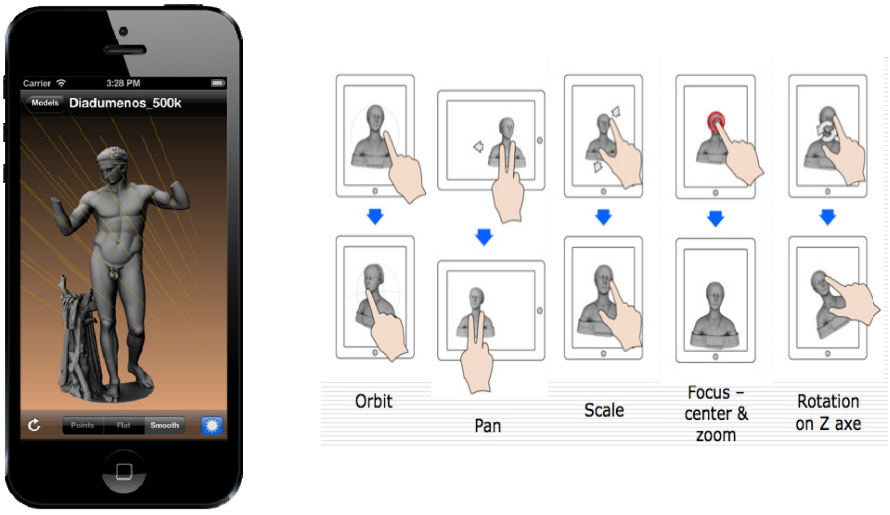

Fig. 4. MeshLab on iOS (on the left) and the touch-based interaction adopted to manipulate the virtual object

A corresponding research project was started at CNR-ISTI at the end of 2011, aimed at developing a visualizer for 3D meshes that should run on smartphones or tablets. We planned to move to those platforms just the visualization component of MeshLab. This tool has been released for both the Apple iOS (http://www.meshpad.org/) and the Android (https://play.google.com/store/apps/details?id=it.isticnr.meshlab) platforms. The goal was to implement a 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 1M - 2M faces. For a more efficient rendering of larger models, we recently ported the Nexus approach also on this platform (the corresponding update for the app will be released in the third quarter of 2013).

3 Manipulation and Interaction

Interaction with CH models is a very complex task, since this specific application domain requires several different access modalities and presents some associated open issues [15], since we need new approaches to support: efficient and easy *manipulation* and *visual inspection* (single object in focus) [16]; efficient and easy *navigation* of large scenes (e.g. architectures) [17]; finally, we should design and experiment with *natural* or *disappearing interfaces* (gesture-based, tracking via Kinect-like devices, etc.).

So far we have been focusing mostly on the first class of interaction techniques, giving priority to the case where the user has to interact with just a single object at a time, and where this single object is inspected by freely manipulating it. In this context, we have worked so far on two research lines: the design of a multi-touch interface for this type of visual manipulation task and the design of constrained manipulations for 3D browsers.

The result of the first research line was the interface implemented for the *MeshLab for iOS* tool (Figure 4), an app available for free on the Apple Store (http://www.meshpad.org/). Conventionally, the manipulation-style interaction is performed with interaction devices providing input with two degree of freedom, like the mouse. The core problem is how to map this limited interaction to 3D transformations, which have at least six degrees of freedom. More importantly, the interaction approach should be intuitive and natural for the user, even with limited input capabilities. With the *multi-touch interaction* approaches provided on current mobile platforms, people use fingers and expect to perform the interaction in the same way they would do for everyday tasks. That is because touch technology allows directly operating on the screen, producing a feeling of naturalness that must be reflected in the interaction technique. For this reason, when designing a multitouch-enabled application, an easy and intuitive interaction scheme is a must. Users want to open the application and start using it with little or no training, also in the case of 3D applications.

Starting from the well known *virtual trackball* [18] approach, our design for the *MeshLab for iOS* app required to replace the well-known *mouse-based interface* with

a more modern and intuitive *touch-based* interaction. Focusing on the single object inspection task allows us to exploit at best the directness supported by the touch-screen interaction. A set of interaction gestures was designed and implemented (see Figure 4): 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; 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.

The touch-based interfaces are becoming predominant also in the web-browsing environment; for this reason, also for the web-based visualization schemes we had to consider the implication of this change. While most of the advantages of the multi-touch interface are filtered by the browser, it is nevertheless true that an interface designed for a *mouse plus keyboard* interaction will perform poorly on a touch-only device. For example, when designing the interface of a WebGL/SpiderGL component, it is necessary to consider that the user might not have a keyboard available (and thus we should provide an alternative for modifier keys), that the visual elements (like buttons) used in the interaction should have at least the size of a finger and a decent spacing, and that some interaction methods (like the double click/tap) are easier than other (the right click).

In terms of usability, the results of some preliminary informal tests of *MeshLab for iOS* performed with CH users (scholars, students or restorers) produced an excellent evaluation. We experimented that users become able to manipulate a virtual object much faster with the touch-based approach than with the usual mouse-based interface provided by the desktop version of MeshLab.

The second research line mentioned above was aimed at the design of specific constrained interactions rules, to be used in all those cases where we think the user should not be allowed to perform generic manipulations. For example, if our artwork is a statue, it could be useful to constrain the manipulation by forcing the virtual statue to keep its vertical orientation (forbidding to rotate the head below the body). This has been implemented, for example, in the browser designed to inspect the Ruthwell Cross (see Figure 5 and [22]), where the user is forced to keep the cross in its vertical disposition. In this case, visual inspection is structured in a cylindrical fashion: right and left mouse drags produce a rotation around the object, while the up and down drags shift the cross vertically along its axis. The choice of a constrained navigation, while somehow limiting the possible viewpoints, greatly simplifies the visualization of an object by providing an interaction that is tailored to its shape. Therefore, we can envision that a generic browser could be defined by supporting different constrained interaction rules, each one best fitting some specific type of object: nearly planar for such artifacts as bas-reliefs; turntable or simplified trackballs for standard 3D objects that we might be interested in inspecting from any side;

mostly linearly-constrained for objects that have a decoration that follows a specific path (e.g. the Trajan column, where one could be interested in following a spiral navigation path that follows the carved decoration); etc. Selecting the proper interaction rule could be a task for the multimedia designer of the specific multimedia presentation, or it could be left accessible to users as an available option.

4 Integration of Different Media – The 3D Model as the Spatial Index to Knowledge

In CH applications, 3D models are major assets to:

- Document an artwork
- Assess the conservation status
- Present the status before and after a restoration
- Disseminate to the large public

Any project, being it devoted to the study of an artwork or to its restoration, produces an incredible corpus of data. This includes historical documents, texts, natural images, and the results of scientific investigations (that are encoded, again, with texts, images or graphs). This complex pool of **data associated to / interlinked with** the 3D model is usually returned and archived as a set of disconnected multimedia documents, usually stored in different places. Those data layers represent the knowledge over a given artwork: retrieving them and establishing all the required connections between those data is a complex, time consuming and valuable work that is usually lost at the end of a project (since the final result is still usually a written report or a book). Ideally, those data should be preserved, enriched by the associated metadata, and access should be granted to all scholars/students/amateurs. Any technological tool that will support the open access to and the easy delivery of those knowledge layers will make a major contribution to a more modern management policy of CH knowledge.

In this framework, the *digital 3D model* can be used to build up **spatial indexes** and to work as the **supporting media** to present other types and sources of information. We have seen that we have new and sufficiently consolidated solutions for distributing and using 3D data on the web (Section 2). Visualization of high-quality 3D models is possible but the required features are not just limited to the interactive visualization / navigation; conversely, we need authoring tools able to enrich the 3D model with hotspots, selection of views and links to other multimedia assets (images, text, graphics, video, audio, etc.) [19]. For this purpose, we must go beyond basic visualization of 3D models and build applications that link the 3D model to the other media, supporting the construction of web presentations that allow an interactive, integrated and efficient access to an entire corpus of knowledge.

A first step in this direction could be to design web pages where the 3D model stores a number of links on its surface, each one pointing to a specific chunk of information that can be activated and visualized with a simple click [20, 21]. The Community Presenter tools (Sect. 2.3) already supports the construction of multimedia applications where the 3D model could play the role of a spatial index to

other type of information elements, by adding hot spots and hypermedia links to other information tokens (text, images, videos, etc.).

A more sophisticated result could be obtained by designing a system able to interconnect different media and to provide a complex network of cross-links and relations. A first attempt on this path has been done recently with the design of a system able to provide a strict correlation between *text* and *3D models* [22]. Many CH artworks are very complex and require long and structured textual descriptions to tell their story, to explain the artistic value and the iconographic content. We usually have to present a story that is narrated by means of the sculpted decorations (in the case of statues, bas-reliefs or even buildings). We cannot just focus on the text or on the visual channel: we should have both media available at the same time, with a tight synchronization in the navigation over each media. Therefore, our goal was to design a modular, extensible framework for the presentation and navigation of interconnected 3D models and textual information, where the user can explore the content in a completely free fashion, or could refer to a set of reference points that link each portion of the textual description to the corresponding ideal view over the 3D model (and vice versa). Some images of the prototype of this system are presented in Figure 5.

The Ruthwell Cross (http://en.wikipedia.org/wiki/Ruthwell_Cross) is the artwork selected as test bed for the design and assessment of our browser. The Ruthwell Cross is an Anglo-Saxon stone cross that presents a number of issues and is thus ideal as a test bed since:

1. It is a large artwork (5.5 meters high) with a lot of carved details on its surface that depict Christ and several other figures;
2. The stone surface is carved, irregular and highly degraded in several regions;
3. The carved figures and decorations have an important symbolic and religious meaning that requires textual descriptions to explain that content to non-experts;
4. Runic inscriptions are carved on the side of several of the carved figures, and require transcriptions and explanations.

Points 1 and 2 are issues that require the adoption of a browser able to present all the richness of a high-resolution model (the full resolution model of the Ruthwell Cross produced with 3D scanning is described by 122 M triangles). Points 3 and 4 suggest the combined use of visual and text media in the presentation of the artwork.

The basic idea is to use different representations of an artwork (possibly including different media), each one annotated, in the sense that each specific point of of the surface of the artwork is associated to a unique ID. If we have several *"models"* representing the artwork (for example a 3D model, an image, a text), each ID can be found in some or all those representations, thus providing a "connection point" which can be used by the user to move seamlessly from one media to another one. Our browsing system therefore should provide a series of viewers, which are HTML/JavaScript objects, each one able to:

- Visualize a specific kind of media/dataset;
- Show the available connection points (and let the user select each of them);
- Move the visualization focus to one specific connection point, according to the request of any other viewer (to synchronize the visualization of the different media).

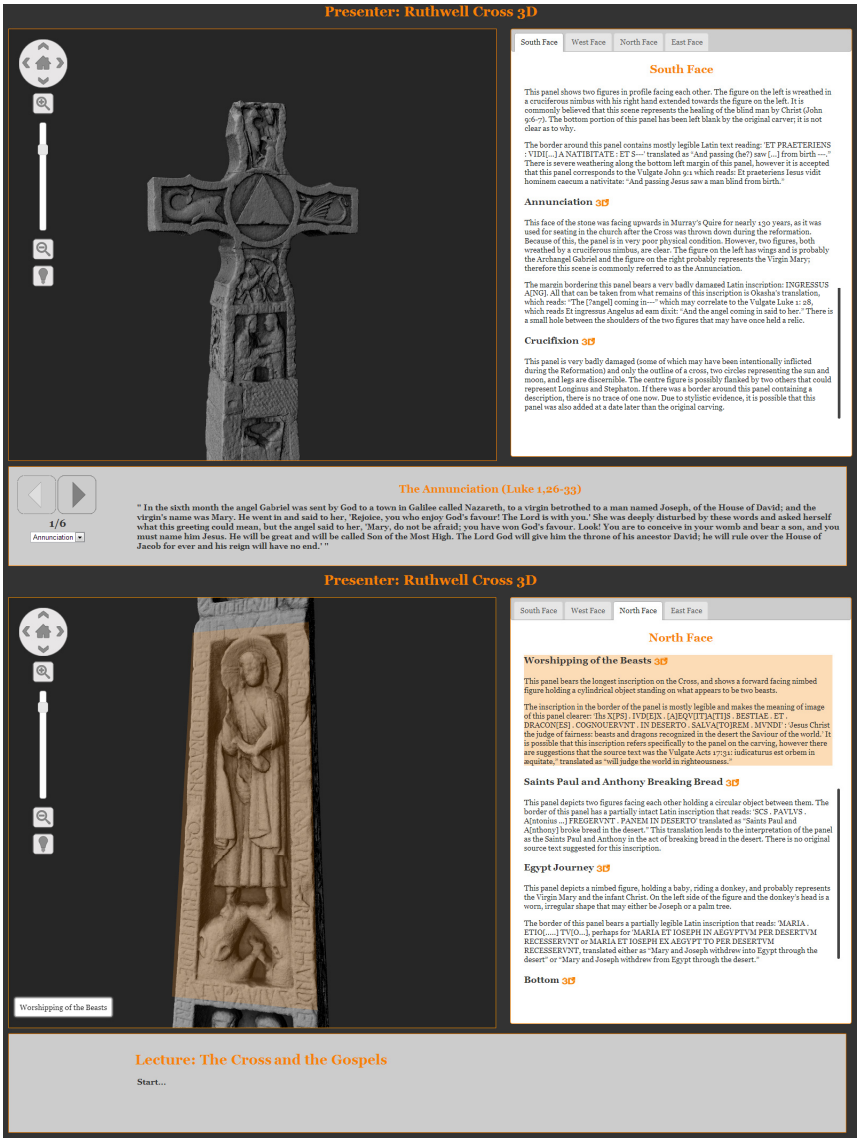

Fig. 5. Two snapshots from our web-based presentation systems, developed on top of Nexus and SpiderGL, that allows to present in an integrated manner a complex artwork, the Ruthwell Cross, by means of interrelated textual descriptions and 3D visualization

The underlying structure needed to synchronize the visualization is basically a dictionary containing the IDs and a map of the annotated datasets, that will act as a synchronization agent. The final user will be able to browse each representation in its specific viewer and, when selecting one of the connection points, synchronize all the

other viewers in the page, to continue the exploration over another media or just to have in foreground the point of interest.

When designing a visualization web page, the author can assemble multiple viewers in the page, to cope with the needs of the project. The various datasets are fed into the system using XML configuration files; with such a separation between the visualization code and the data, it is possible to build different web-based visualization schemes without having to modify JavaScript code, but just authoring the XML content, a task suitable also for non programmers.

While our prototype [22] is at the moment limited to textual and 3D representations, it is based on a general concept of free and synchronized exploration of an artwork across all its representations and it can be further extended to other media (a set of photos, videos, technical drawings and surveys).

This system has been developed with the idea of using it as an educational tool: a sort of interactive multimedia textbook, usable in a class to present a specific artifact or, later on, to support the student/scholar when studying the object. To strengthen this idea, we added to the system another component whose purpose is to present a specific, annotated, exploration path of the available datasets that uses the available connection points (see the grey bar on the bottom of the window in Figure 5). We believe that this component may be extremely useful for a teacher when preparing and presenting a lesson or for a student when preparing and presenting an essay.

5 Conclusions

The delivery of informative rich and high resolution three-dimensional content on the web and on mobile devices has been an open issue for a long time, due to the lack of standards, practices and because of hardware limitations. The improvements and novelties that appeared in the last few years have finally opened new perspectives for a full integration of 3D contents in the everyday life of users and in some important professional contexts, like Cultural Heritage.

We have presented a number of new technologies, encompassing efficient multiresolution representation and rendering solutions, for both local and web-based visual presentation, together with technologies for an improved interaction with those multimedia models, covering also the domain of touch-based platforms. Finally, we have briefly touched the integration issue, demonstrating how the multimedia technologies could be extended to support the capability of interlinking and integrating different media.

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 and we are educated on artistic themes is impressive. Those technologies could bring a revolution in the way people document, communicate or teach arts. Even if technologies are now starting to be mature, there have been limited efforts so far with respect to large scale deployment and assessment efforts. Two current EU projects (the NoE “V-MUST” - http://www.v-must.net/ - and the infrastructure project “ARIADNE” - http://www.ariadne-infrastructure.eu/) will give

us an ideal framework for producing cutting edge experiments for a larger scale dissemination and assessments of these integrated technologies.

Moreover, other potential directions of work, exploiting new technologies like natural interfaces and immersive 3D visualization, could provide additional breakthroughs concerning user immersion and usability.

Acknowledgments. This work has been partially partially founded by the European Projects Ariadne (FP7–INFRA–2012–1–313193) and NoE V-MUST.Net (FP7 *Grant Agreement no. 270404*).

References

1. Callieri, M., Cignoni, P., Corsini, M., Scopigno, R.: Masked photo blending: mapping dense photographic dataset on high-resolution sampled 3D models. Computers and Graphics 32(4), 464–473 (1981)
2. Dellepiane, M., Marroquim, R., Callieri, M., Cignoni, P., Scopigno, R.: Flow-Based Local Optimization for Image-to-Geometry Projection. IEEE Transactions onVisualization and Computer Graphics 18(3), 463–474 (2012)
3. Lensch, H.P.A., Kautz, J., Goesele, M., Heidrich, W., Seidel, H.-P.: Image-based reconstruction of spatial appearance and geometric detail. ACM Transaction on Graphics 22, 234–257 (2003)
4. Scopigno, R., Callieri, M., Cignoni, P., Corsini, M., Dellepiane, M., Ponchio, F., Ranzuglia, G.: 3D models for Cultural Heritage: beyond plain visualization. IEEE Computer 44(7), 48–55 (2011)
5. Corsini, M., Dellepiane, M., Dercks, U., Ponchio, F., Keultjes, D., Marinello, A., Sigismondi, R., Scopigno, R., Wolf, G.: CENOBIUM - Putting Together the Romanesque Cloister Capitals of the mediterranean region. B.A.R. - British Archaeological Reports International Series, vol. 2118, pp. 189–194. Archaeopress (2010)
6. Kim, M.H., Rushmeier, H.E., French, J., Passeri, I.: Developing Open-Source Software for Art Conservators. In: VAST 2012 Proc., Eurographics, pp. 97–104 (2012)
7. Borgeat, L., Godin, G., Blais, F., Massicotte, P., Lahanier, C.: GoLD: interactive display of huge colored and textured models. ACM Trans. Graph. 24(3), 869–877 (2005)
8. Wimmer, M., Scheiblauer, C.: Instant points: fast rendering of unprocessed point clouds. In: Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics, SPBG 2006 (2006)
9. Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Ponchio, F., Scopigno, R.: Adaptive tetrapuzzles: Efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. ACM Trans. on Graphics (SIGGRAPH 2004) 23(3), 796–803 (2004)
10. Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Ponchio, F., Scopigno, R.: Batched multi triangulation. In: IEEE Visualization 2005, pp. 27–35 (2005)
11. Khronos Group. WebGL - OpenGL ES 2.0 for the Web (2009)
12. Behr, J., Eschler, P., Jung, Y., Zollner, M.: X3DOM: a DOM-based HTML5/X3D integration model. In: Proceedings of the 14th International Conference on 3D Web Technology (Web3D 2009), pp. 127–135 (2009)

13. Di Benedetto, M., Ponchio, F., Ganovelli, F., Scopigno, R.: SpiderGL: A JavaScript 3D Graphics Library for Next-Generation WWW. In: 15th Conference on 3D Web Technology, Web3D Consortium, ACM Web3D 2010, pp. 165–174 (2010)
14. Palma, G., Siotto, E., Proesmans, M., Baldassari, M., Baracchini, C., Batino, S., Scopigno, R.: Telling The Story of Ancient Coins By Means Of Interactive RTI Images Visualization. In: CAA 2012 Proceedings of the 40th Conference in Computer Applications and Quantitative Methods in Archaeology, Southampton, United Kingdom, March 26-30 (2012)
15. Christie, M., Olivier, P., Normand, J.-M.: Camera control in computer graphics. Computer Graphics Forum 27(8), 2197–2218 (2008)
16. Khan, A., Mordatch, I., Fitzmaurice, G., Matejka, J., Kurtenbach, G.: ViewCube: a 3D orientation indicator and controller. In: Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games (I3D 2008), pp. 17–25. ACM, New York (2008)
17. Fitzmaurice, G., Matejka, J., Mordatch, I., Khan, A., Kurtenbach, G.: Safe 3D navigation. In: Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games (I3D 2008), pp. 7–15. ACM, New York (2008)
18. Chen, M., Mountford, S.J., Sellen, A.: A study in interactive 3D rotation using 2D control devices. In: Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1988, pp. 121–129. ACM, New York (1988)
19. Faraday, P., Sutcliffe, A.: Designing effective multimedia presentations. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 1997), pp. 272–278. ACM, New York (1997)
20. Jankowski, J., Samp, K., Irzynska, I., Jozwowicz, M., Decker, S.: Integrating Text with Video and 3D Graphics: The Effects of Text Drawing Styles on Text Readability. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2010), pp. 1321–1330. ACM, New York (2010)
21. Polys, N.F., Bowman, D.A., North, C.: The role of Depth and Gestalt cues in information-rich virtual environments. Int. J. Hum.-Comput. Stud. 69(1-2), 30–51 (2011)
22. Callieri, M., Leoni, C., Dellepiane, M., Scopigno, R.: Artworks narrating a story: a modular framework for the integrated presentation of three-dimensional and textual contents. In: Proceedings of the ACM 18th International Conference on 3D Web Technology, Web3D 2013 (2013) (in press)
23. Koller, D., Frischer, B., Humphreys, G.: Research challenges for digital archives of 3D cultural heritage models. ACM J. Computing and Cult. Herit. 2(3), Article 7 (January 2010)
24. Patel, M., White, M., Mourkoussis, N., Walczak, K., Wojciechowski, R., Chmielewski, J.: Metadata requirements for digital museum environments. International Journal on Digital Libraries 5(3), 179–192 (2005)