INTERACTIVE ONLINE VISUALIZATION OF COMPLEX 3D GEOMETRIES

Argyro-Maria Boutsi, Charalabos Ioannidis, Sofia Soile

Laboratory of Photogrammetry, School of Rural & Surveying Engineering, National Technical University of Athens, Greece;
iro_mpoutsi@outlook.com, cioannid@survey.ntua.gr, ssoile@survey.ntua.gr

Commission II

KEY WORDS: 3D visualization, 3DHOP, three.js, navigation, Cultural Heritage, WebGL

ABSTRACT:

In the last decade 3D datasets of the Cultural Heritage field have become extremely rich and high detailed due to the evolution of the technologies they derive from. However, their online deployment, both for scientific and general public purposes is usually deficient in user interaction and multimedia integration. A single solution that efficiently addresses these issues is presented in this paper. The developed framework provides an interactive and lightweight visualization of high-resolution 3D models in a web browser. It is based on 3D Heritage Online Presenter (3DHOP) and Three.js library, implemented on top of WebGL API. 3DHOP capabilities are fully exploited and enhanced with new, high level functionalities. The approach is especially suited to complex geometry and it is adapted to archaeological and architectural environments. Thus, the multi-dimensional documentation of the archaeological site of Meteora, in central Greece is chosen as the case study. Various navigation paradigms are implemented and the data structure is enriched with the incorporation of multiple 3D model viewers. Furthermore, a metadata repository, comprises ortho-images, photographic documentation, video and text, is accessed straight forward through the inspection of the main 3D scene of Meteora by a system of interconnections.

1. INTRODUCTION

Online 3D models are nowadays widespread and vary from wireframe models and digital sculpting to high-resolution digitized geometries. Focusing on geometric quality, triangulated meshes, point clouds and textured models are some of the products of several techniques that have been developed and involved in the last years (Structure from Motion, multi-view stereo, laser scanning, etc.) Even though the acquisition of these high-quality data is versatile, their web-based visualization remains quite a challenge. Rendering time, lossless transmission, compatibility and interaction design constitute the main issues that the current technologies should deal with.

Concerning the benefits, limitations and applicability to Cultural Heritage field, the high resolution of the produced 3D models is perceived both as a potential and a problem. Data sizes are extremely big for streaming, as for example high-density sampled meshes created from laser scanning. So, internet users cannot distinguish the geometric accuracy and evaluate the high quality of the prototypes if they cannot even access them. Taking into account the low-end platforms of non-specialized audience, it seems that it is impossible to convey correct information to them.

Hopefully, a great number of techniques able to produce controlled granularity and surface simplification can compensate for the possible slow network performances or the limited computing resources. These techniques are divided into three classes (Ponchio, 2008):

- filtering: techniques to discard all the information which is not relevant for the current scene,
- data management: techniques to retrieve the selected data and move it to the graphic hardware in the most efficient manner,
- rendering: techniques to fully exploit the computational power of the graphic hardware.

Rendering methods are recommended for large high-resolution models because of lossless transmission and protection of the original data. Multiresolution schemes split the geometry in chunks with different levels of details and a gradually refinement occurs on load depending on the view. The fact that only the needed information for the visualization is transferred minimizes the latency and makes it difficult to recreate the prototype model.

The presence of high-resolution 3D models in online repositories, eventually exploiting any of the previous techniques, does not account for the lack of supportive 2D metadata. Cultural Heritage documentation concerns professionals from different knowledge areas, thus manifold content, various formats and data structures should be demonstrated. To serve dissemination of knowledge or educational purposes, the relative information should be managed at a multi-scale level. For example, georeferenced ortho-images engage the attention of archaeologists and engineers, whereas 3D models appeal to the general public. A link between these data could facilitate both user groups.

The contribution presented in this paper is built upon these observations. We have developed novel interactive tools for the visualization of complex 3D geometries, such as rugged terrains, cultural landscapes and densely built-up areas. The framework can be accessed as a common web page and it is built upon 3DHOP (Potenziani et al., 2015), WebGL API (Khronos Group, 2009) and Three.js library (Cabello, 2010). It aims to address the real needs of a simple user and the intended purposes of a researcher. The display of large heterogeneous datasets of 3D models, regardless of resolution and analysis is supported. Moreover, geometric, topological and learning information can derive from the virtual 3D scenes in an interactive way.

The paper is structured as follows: the status of technological methods for online 3D publishing is presented in the next section and the architecture along with the setup of the proposed system is exposed in section 3. An elaboration on the developments and new features of the framework is made in section 4 and section 5 details the case of study, data acquisition and processing. The evaluation of the project and the conclusions are presented in sections 6 and 7.

2. RELATED WORK

Early approaches proposed for the delivery of the 3D content through viewers or browsers combined interfaces like OpenGL, Direct 3D or Java 3D with formats as XML, VRML, X3D and MPEG.

In the early 90's, 3D visualizations on browser were out of scope. The complexity and the lack of standardization of 3D data discouraged their integration to the web as the rest multimedia content. Hence, the first attempts were delegated to external software components, such as Java applets (Sun Microsystems) and ActiveX controls. However, the demand of 3D accelerated gaming led to the appearance of software with supportive cross-hardware graphics library, OpenGL. The first version of OpenGL API was released on 1992 by Silicon Graphics Inc. (SGI) and since 2006 it is controlled by Khronos Group. The main competitor of it is DirectX, released by Microsoft in 1995. OpenGL exploits the maximum of the capabilities of the graphic card and its 4.0 version pipeline, similar to Direct3D 11, uses tessellation to subdivide data. While the last one is used mostly in computer games, OpenGL dominancy in the professional market arises from its performance. Worth mentioned is Glide API, developed by 3dfx Interactive. Although it gained a lot of popularity in the late 1990s as a cross-platform 3D graphics API, it ceased operations and 3dfx's assets were purchased by NVidia.

The former web standard for interactive 3D graphics was VRML defined by the Web 3D Consortium as a text file format in 1997. It was replaced by X3D in 2004, an open ISO standard with XML encoding and binary data compression. It was designated as a unified specification of online 3D content. Despite the definition of event handlers and interaction utilities, it is designed to render with specific software, such as browser plugins or X3DOM (Behr et al., 2009). Guarnieri et al. (2009) introduced an online interactive viewer that relies on X3D data format, PHP and PostgreSQL database for cultural Heritage 3D models. The visualization occurs with Octaga plug-in for browsers and the portal includes not only 3D models of artifacts and 3D assets but a number of relative attributes and metadata.

The need of intuitive, easy and independent of any technical environment visualization resulted in the appearance of new plug-in free solutions, without explicit installation procedures. The successor of X3D, X3DOM constitutes an open source JavaScript framework, structured on a declarative method of scripting. It performs both as a server side and client side system; it is based on WebGL with XML file format. Opposite to X3DOM, XML3D (Sons et al., 2010) leverages the existing HTML properties in order to configure the 3D scene. However, the declarative markup of X3DOM and XML3D causes difficulties when loading very large models due to the size of the geometry data that are parsing.

The definition of figures in a browser grew with the release of the markup language HTML5 and the canvas element embedded into its content. In 2012, even Adobe stopped all development of its plugin Flash, and focused on HTML5 for browser content. On top of HTML5 technology, O3D plugin (Google, 2009) of C/C++ assigned the permission of Direct3D or OpenGL to operate with the GPU to a JavaScript API.

By 2009, there was no standard method of accessing the GPU directly via the browser. A new cross-platform's rendering outperformed both canvas and SVG and proposed that 3D graphics should be a correlative of these web technologies. WebGL standard (Khronos Group, 2009) is a proposed binding of standard OpenGL to Javascript and HTML5 canvas. HTML 5.0 was set as standard language for building websites in October 2014 and since then WebGL is incorporated by all modern web browsers (except iOS mobile devices) bound together by its descriptive semantics. Its advent has ignited the development of various commercial platforms, libraries and academic and institutional frameworks for publishing content on the web.

Some of the libraries of imperative scripting that set up the scenegraph concept are Scene.js (Kay, 2009) and GLGE (Brunt, 2010), whereas SpiderGL (Di Benedetto et al. 2010), WebGLU (DeLillo, 2009) and Three.js constitute simpler declared paradigms. One of the precursors of JavaScript libraries devoted to WebGL is SpiderGL. It provides a series of high-level helpers and structures to facilitate the defining and designing of 3D visualizations and web applications. Leoni et al. (2013) presented a multimedia framework that links textual and spatial information to a highly detailed 3D Cultural Heritage model on top of the SpiderGL library. Different types of data are encoded and interlinked through XML configuration files and the visualization takes place with HTML5 and JavaScript. GLGE library mimics X3DOM and XML3D as regards to the declarative definition of the scene. Three.js is probably the most popular open source javascript library for tailor-made WebGL based 3D viewers. It has the potential to integrate various types of data and 3D models (point clouds, colored vertex meshes, textured meshes). The scene is a synthesis of the mesh, the material, the camera and the light, just like XML3D. A new release is Potree by EcoSynthis, a viewer for point cloud rendering with color per vertex texture information. The online presentation occurs with the help of point splatting techniques, based upon three.js library.

The advent of Flash Player 11 in 2011 came with Stage3D, an Adobe Flash player API for facilitation GPU acceleration of 3D content in Flash applications. It is similar in purpose and design to WebGL. It consists of an API for the manipulation of texture memory, mesh rendering and shading and support system for graphics card. In 2012 Sketch Fab was launched, an upload-and-share service that supports 28 different file formats through WebGL. The 3D models are converted and then reside on the software's server and for substantial storage a monthly fee is required. 3drepo.io (Scully et al., 2015) is also a cloud hosting service leveraging AngularJS and X3DOM. Other WebGL-based applications that use cloud storage are p3D, Sunglass and 3Dsom.

The prevalent serious games engine for high-end online presentations is Unity3D with Unreal to gain ground. Unity3D and its Web player are adopted by all major desktop platform and offer extended simulation capabilities (collision detection, notion of space). It has no competitors in the terms of management of complex geometry and re-use of 3D assets. After its creation, the 3D scene is exported to a custom file format and the Unity web-browser plugin displays it in the browser window. The scene can be imported in several applications, even in mobile platforms. A constrain to its use is the automatic fragmentation when handling high-resolution 3D models. Unreal game engine is embedded in development kits in order to be used as a visualization engine of complex 3D data. Compared to Unity3D, its installation is a bit more technical but it allures the academic community with a HTML5/WebGL-based demo and its free version for non-commercial use. QueryArch3D (Agugiaro et al., 2011) is a Unity implementation for interactive visualization of 3D models of Cultural Heritage field (Maya Archaeological site in Copan, Honuras). Once the geometry is loaded in Unity corresponding attributes are declared in PostgreSQL.

Some relative projects that correlate 3D data with their attributes are cited below. Hyper3D (Kim et al., 2014) is a XML-based software tool that correlates the visualization of 3D surface models and images with text and graphic information. Agata (Soler et al., 2016) is a 3D information system, annotating raster and vector data onto high resolution 3D models. It uses XML data format, multiresolution rendering of 3D models and the information layers are stored in a SQL database. On the basis of 3DHOP, ADS 3D Viewer (Galeazzi, et al, 2016) is a platform for the visualization and analysis of 3D archaeological data derived from the Archaeological Data Service (ADS) repository. Both single 3D models and aggregated data (archaeological statigraphy) are displayed into two independent viewers. Apollonio et al. (2017) designed an information system (IS) to support the restoration of Neptune's Fountain in Bologna on the basis of 3DHOP technology regarding the 3D data.

3. METHODOLOGY AND SETUP

The proposed system depends on 3DHOP, therefore on WebGL and its inherited library, SpiderGL. A typical WebGL application merges HTML5 and JavaScript with GLSL (Graphics Library Shader Language) code and is loaded into the Canvas element.

The SpiderGL and the embedded to HTML page Three.js loader abstract WebGL inner workings and core functionalities and create intuitive elements depending on their architecture.

3.1 Architecture of the framework

Applying client-side architecture does not require a computer network configuration nor a specialized web server or server-side daemon. Currently, the JavaScript files, the HTML page, the Cascading Style Sheets (CSS) files and the multimedia are stored on Apache HTTP local server. These files get accessible on a remote web server by IP. The client sends a request to the server, addressed by localhost. The server is looking for the requested document, transfers the document to the client and the browser is rendering the file information into a readable format. The browser renders the 3D data accelerated by the GPU of the host machine. Server and client communication is bidirectional and rendering process is cyclic. Predefined connection points are marked as links user can follow to browse the datasets. New browser windows open through the URLs navigation and dialog boxes pop up information without redirecting to new pages (Figure 1). The 3D models and graphics are loading in the HTML Canvas element. Because of the independent distribution of resources to the 3DHOP and Three.js datasets, the visualization can be twofold; content can be rendered at the same time (Figure 2).

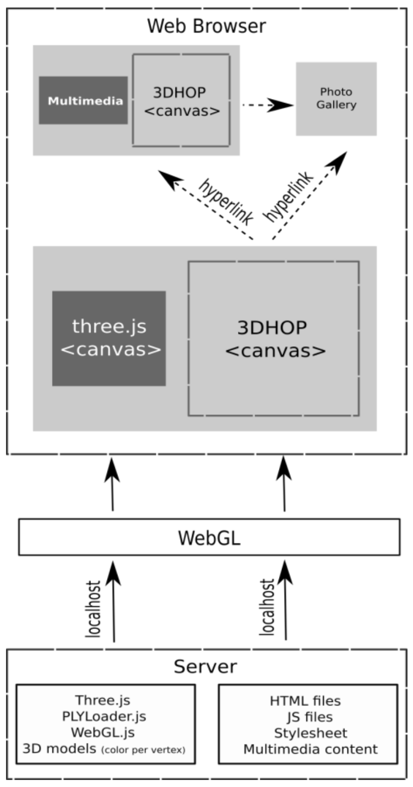
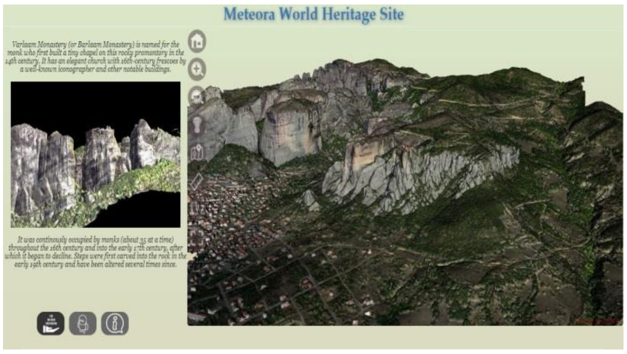


Figure 1. Overview of framework's architecture and operation process



Varlaam Monastery (or Barlaam Monastery) is named for the monk who first built a tiny chapel on this rocky promontory in the 14th century. It has an elegant church with 16th-century frescoes by a well-known iconographer and other notable buildings.

It was continously occupied by monks (about 35 at a time) throughout the 16th century and into the early 17th century, after which it began to decline. Steps were first carved into the rock in the early 19th century and have been altered several times since.

Figure 2. The framework's main page featuring the 3D model of the archaeological site of Meteora and a part of it on a smaller Three.js viewer (left of the window)

3.2 3D Heritage Online Presenter

The selection of 3DHOP as the core software for our framework is due to these characteristics:

- it supports a multiresolution rendering technique and hence, network load and latency are reduced,
- its viewer is an independent HTML element that relies on WebGL standard,
- it supports a declarative way of defining the 3D scene,
- it is based on a multi-thread JavaScript structure with exposed functions and events for the control of the interaction with the rest of the HTML page,
- it has modular structure that facilitates the development of new, specialized components and the tuning of existing ones.

3.3 JavaScript and JQuery

WebGL is a JavaScript API; any operation is executed as fast as the code behind it. Thus, the selection of JavaScript is indisputable. In essence, JavaScript controls the Document Object Model (DOM) and customizes user interface scripting, automation, animation and client-side validation. The JavaScript’s interaction with HTML is handled through the manipulation of the page by coded responses or certain actions (causing buttons to open windows, messages to be displayed etc.) The use of elements is actually carried out by the functions that are called in response to these events, the event handlers. Event handlers are assigned both in JavaScript and in HTML. In JavaScript, the object in question has to be referred and then a function is assigned to the corresponding event handler property. In HTML, an event handler attribute is added to the HTML tag with the appropriate script in the role of a value, such as “click” or “mouseover”. However, the accomplishment of their assignment requires the DOM method addEventListener.

The redundancy of plain-vanilla JavaScript could be removed prioritizing a more concise and obfuscated code. Thus, JQuery, a lightweight JavaScript library, is extensively used. It gives a standard Event API and Selectors API and useful enumeration methods that work across the board in Internet Explorer, Safari, Firefox, and Opera on Windows, Mac and Linux platforms. Its latest version, JQuery 3.3.1, is included to the HTML document by a single link of markup. It patches over certain portions of cross-browser development like the event model. The built-in selection mechanism it provides creates a performance boost by accessing and traversing the DOM once. This fact is equivalent to faster loading and easier dealing of bugs.

3.4 Three.js WebGL Renderer

Three.js is an open-source library/API, devoted to WebGL that simplifies its convoluted code and enriches it with prebuilt and high-end 3D graphics. Apart from its popularity, it is the most complete library from a features standpoint. A real-time 3D visualization is flexible and easily configurable as Three.js is designed to integrate with the DOM. It uses a plug-in system for the generation of the WebGL renderer. After its creation, it is appended to the DOM via the body element with a custom width and height. Scene object is the most important declaration because it serves as a container in which all other graphical objects are placed. Then, the camera is added to the scene with the aid of three dimensional coordinates. The perspective Camera’s parameters are the field of view (FOV), the viewpoint dimensions, and front and back clipping plane values. Before the loading of the geometry, it is essential to define a light to a relative position.

Three.js supports a variety of 3D file formats either in ascii or binary encoding. Our imported 3D models are expressed in PLY format so as to geometric and color information to be handled at once. The mesh object consists of a geometry object (shape) and a material. Each 3D object carries transform information, represented in the properties position, rotation and scale. Their surface attributes correspond to textures, namely JPEG images. An API call loads the texture image asynchronously. When the image is assigned to a texture, it is actually transferred from processor memory (CPU) to a texture memory where every bit is mapped onto the correct position of each face of the mesh.

Standard orbit controls are included in order to achieve interactivity; zooming in or out with the mousewheel, rotation by clicking and dragging with the mouse and panning by right-clicking. The scene is rendered after the configuration of some exposed functions. RequestAnimationFrame() implements the essential run loop and updateProjectionMatrix() prevents toggling camera settings in case of browser resizing.

3.5 Rendering methods

The way 3D content is treated from predefinition to rendering is distinct from viewer to viewer. Three.js library contains a scene graph with all the objects that should be delivered on the web. Our imported 3D models are expressed in PLY format so as geometric and color information to be handled at once. A per-wedge to per-vertex transformation, using MeshLab software, results in a better representation of physical materials and real-time shadows. When browser calls the API, the scene graph is updated without the demand of drawing the complete scene on every frame. It is the library’s decision when the actual drawing should be done (Anyuru, 2012). Relying on a dissimilar method, client performs the view-dependent visualization on 3DHOP’s canvas. The corresponding 3D models are converted from a single-resolution to a multi-resolution format using Nexus library. Nexus operates as view-dependent level-of-detail rendering algorithms. The subdivision scheme, based on clusters of different levels of detail boosts GPU streaming performance. Once the 3D multi-resolution data are delivered to the client for rendering, a gradually loading occurs depending on the view. The visible parts of the object are refined by preference and more data are loaded when zooming in. The progressively refinement countervails the delay in transmission and the large traffic volumes.

4. IMPLEMENTATION

Our framework supports high-quality visual appearance complying with a precise geometric representation and has been designed as an online repository of 3D and 2D data. In terms of Programming, the modular structure of its core, 3DHOP endorses modifications with general applicability. Some of the innate tools are used as they are set by default; others are deployed for further elaboration while others are disconnected. New, advanced components are developed and integrated to the codebase to attain high configurability level that it needs to cope with the diversity of the cultural heritage models. The online presentation is enhanced by the incorporation of another web visualization system for multiple 3D scenes in the same browser window.

4.1 Case study: Meteora

The implementation is done on top of the 3D documentation of the archaeological site of Meteora, conducted by the Laboratory of Photogrammetry, National Technical University of Athens.

4.1.1 General information about Meteora

Meteora is located in central Greece (Thessaly) and occupies a large part of the Antichasia Mountains and the region of Kalambaka. The rock pillars of Meteora, rising over 400 meters above ground level, have a variety of spectacular shapes. The Meteora rock pinnacles with the surroundings are designated by NATURA 2000 as protected area. Besides the unique geomorphological landscape, the archaeological site of Meteora is included in the UNESCO world heritage list in 1988 as it constitutes one of the largest built complexes of Eastern Orthodox Church. The six (of an original twenty four by the late 16th century) monasteries are built on immense natural pillars and hill-like rounded boulders and were established in the 14th century. Apart from the well-known monasteries, a number of hermitages and chapels appeal to the visitors.

4.1.2 Data acquisition and processing

The 3D model of Meteora covers an area of 10 km^2 and includes the 6 km^2 area of the designated archaeological site. Considering the great extend and the morphological complexity of the natural environment, data acquisition was carried out with manned aerial mapping. A total of 2500 vertical and oblique images were captured. The orientation of images using the Structure from Motion (SfM) method and the transformation of point cloud data into a surface of triangular mesh were implemented in Agisoft PhotoScan software. The mesh was improved by filling holes, optimizing the triangles and reducing data locally where needed using Geomagic software. Finally, it was textured using photographs.

The 3D model of Pixari, a natural pillar with hermitages (112 meters in length and height of 155 meters) is also incorporated in the framework. The data acquisition was carried out in a combination of image-based methods using aerial and terrestrial images and range-based technologies (Terrestrial Laser Scanning). Two UAV were used; an Octocopter and the Dji Phadom Pro 4 with an embedded camera of 4096 x 2160 pixels analysis. The images that were shot from the UAV flights captured the inaccessible parts of the pillar. The scans acquired during the survey were aligned and merged to generate a dense point cloud using Geomagic software. The cloud was then filtrated and triangulated to derive the mesh. Photographic texturing was applied in the environment of Agisoft PhotoScan software. The final model of Pixari is composed of 10 million triangles with color mapping using a vertex encoding. Orthoimages and orthophoto-mosaics were also generated.

4.2 Existing toolset of 3DHOP

The default toolset of 3DHOP provides a number of interactive functionalities, like camera and lighting control, a measurement suite, the capability to get orthogonal sections of the 3D model and the coordinates of any point of its surface. Besides, the possibility of a freely exploration of the scene, the framework has the potential of focusing the attention of the user to specific points of interest.

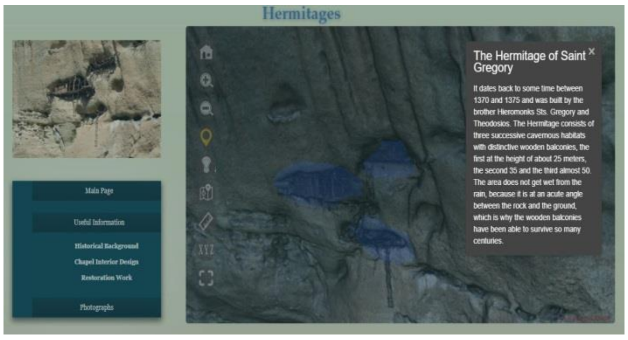
The fundamental tool that carries out the movement of the object in front of the camera is the entity called trackball. Depending on the nature and the shape of the object as well as the visualization needs, four types of trackballs with different behavior are provided. The starting view distance from the object, the angles of rotation around the three axis and the events like zoom and pan assigned to the mouse buttons are easily editable parameters. 3DHOP has also the potential to automate the trackball using JavaScript functions. By calling them from the HTML elements of the page, the position of the camera framing the scene is moved (instantaneously or using a smooth animation). An interest feature, that it is evolved in the presented framework, is clickable geometry. A way to create geometrical hotspots and a series of events to detect user clicks on 3D models and hotspots is available. The toolset is also equipped with the option to section in real-time the 3D models in the scene as well visibility and light direction controls. Finally, the measurement suite and the point-picking tool are extremely useful for georeferenced data. They are based on the principle of spatial measurements as the original coordinates of any point of the object are preserved. The first one returns the Euclidean distance of two points (expressed in meters) when the user picks them. The other gives the coordinates of a point when it is picked on the surface of the 3D model.

4.3 Multimedia incorporation

Our platform constitutes a web-based repository of 3D models of different scale, topological complexity and level of detail. Their high accuracy ensures the quality of the visualization and their unique nature intrigues academia and general public. However, it cannot be perceived as a complete and pluralistic solution if lacks multimedia content. The approach goes beyond the consolidated 2D and 3D graphics, presenting the different media in a coordinated and integrated manner.

First step is the creation of a multi-level, responsive menu with hover reveal, sliding effect with different types of information. A more intriguing approach, rather than a common navigation bar could be an interactive metadata schema. Thus, the interconnection of the 3D content with its corresponding 2D media is achieved by taking advantage of the ability of clickable geometry. Spots of the scene with a particular interest can be highlighted in order to elicit the attention of the user. If a spot is selected, a smooth animation occurs and the camera frames it. The delegation of multiple exposed functions to multiple hotspots could lead to convoluting coding. For complex sets of conditions and operators nested really deeply, the most efficient construct is a "switch" statement. When a spot is clicked, its expression is compared with the value of each case and the first match dictates the beginning of the execution of the code in the body of the statement. There is also a default case which is run when no other case is matched. The navigation bar on the right side of the window toggles dynamically to the selection of the spot. Only the corresponding media of its categories/ subcategories are delivered to the user. The text pops up in modal that perfectly coordinates with the visualization (Figure 3). The jQuery UI plugin provides pseudo-pop-up windows, created using purely markup, CSS and script. Upon calling the dialog() method, the modal is automatically opened. By default, it is allowed to control the dynamic interaction by passing options to the dialog() method; the draggable option is set to true to enable dragging the dialog, and the resizable option is set to false to disable resizing.

The navigation bar comprises links to photo galleries with photographic documentation of the exterior and interior of the landmark, metric data (orthoimages), historic evidence images and artworks. The photographs can be downloaded or shared to the social media in full analysis. Interlinks are also provided for redirection to the main scene of the platform and hyperlinks navigate the user to external sources.



The Hermitage of Saint Gregory

It dates back to some time between 1370 and 1375 and was built by the brother Hieromonks Sts. Gregory and Theodosios. The Hermitage consists of three successive cavernous habitats with distinctive wooden balconies, the first at the height of about 25 meters, the second 35 and the third almost 50. The area does not get wet from the rain, because it is at an acute angle between the rock and the ground, which is why the wooden balconies have been able to survive so many centuries.

Figure 3. By selecting the highlighted spot, the view direction is rotated automatically to face the hermitage of Saint Gregory (Meteora) and the user can obtain information about the historical background.

4.4 Development of new navigation tools

It's crucial that the 3D model of the archaeological site of Meteora depicts its morphological complexity and its world heritage status to the public. In cases when such a large number of landmarks are involved over an area of 10 km^2 a completely free navigation via orbit controls may be misleading and not realistic. For this reason, two types of navigation are developing; an automatic narrative virtual flight and an exploratory walk-through.

4.4.1 Automatic virtual flight

An important entry in the 3D interface is the supervised navigation, a lightweight guided tour that provides seamless transitions between different viewpoints. Once the feature is enabled, a walkthrough is stimulated and the user perceives the virtual scenes from elevated vantage points and at eye height via translations and rotations of the 3D environment. This new experience allows the user to overcome the problem of non-visible offering in the same time the perception of shapes and occupied spaces. Simultaneously, location-relevant information are retrieved and presented automatically into a text field above the scene. The synchronization of camera position to the corresponding description contributes to the critical awareness of the prominent features (Figure 4).

The system designed is lightweight without complex pre-processing operations. Its setup is based on a timed Javascript function which invokes the several animations of the camera framing the scene on all desirable positions, one after another. "setTimeout" is more suited to this operation than "setInterval", which is not bound by the operation of the function it calls. It is used to obviate the risk of having successive calls interfere with each other, especially when long processing time is involved. Each of point of view comprises of a vector containing the parameters of the current state of the trackball, as they are returned through the console "log()" function. All of those configurations are editable and reusable so that a developer can tailor the navigation to suit his/her needs. The logic behind the supplementary supertitles at runtime with the guided tour is also simple. Instead of the replacement and manipulation of the text within a string by back-references, a callback function is used. The "setInterval" automatically repeats the execution of the text swap at ongoing intervals of the time period. The list of the context is stored in an array, and each time the function is called, a variable is updated in order to contain the current index being used. The variables are concatenated into a jQuery selector. Each description fades to the specified opacity and speed (time in milliseconds). The complete configuration creates a narrative experience by delineating the name of the landmark shown, historical background and information of its current state.

4.4.2 Exploratory investigation of the 3D scene

A heterogeneous dataset of the cultural heritage is difficult to be parsed functionally in order to communicate the spatial interconnections to the simple users. An option would be the connection of all available information with the corresponding parts of 3D models. The attachment of the multimedia to a dynamically 3D scene is optimal when user manipulates it in real time. A novel feature, inspired by annotation systems and digital archiving, provides an interactive real-time inspection of the 3D model. While freely exploring the 3D scene with the trackball, the user gets details on parts of interest and gains access to an ancillary documentation of the archaeological site.

Nothing is highlighted or marked; the user moves the cursor at his/her own discretion walking through the model. When the cursor indicates a landmark or a special feature, it changes to a finger pointer. If the user clicks on this spot, a dialog box is shown. Unlike conventional labeling systems, it is not about text notes or trivial attached labels. The text description contains interlinks that browse the user to the pages of the framework with the relative 2D and 3D data. The tool also achieves the emulation of the views a user will see during the navigation.

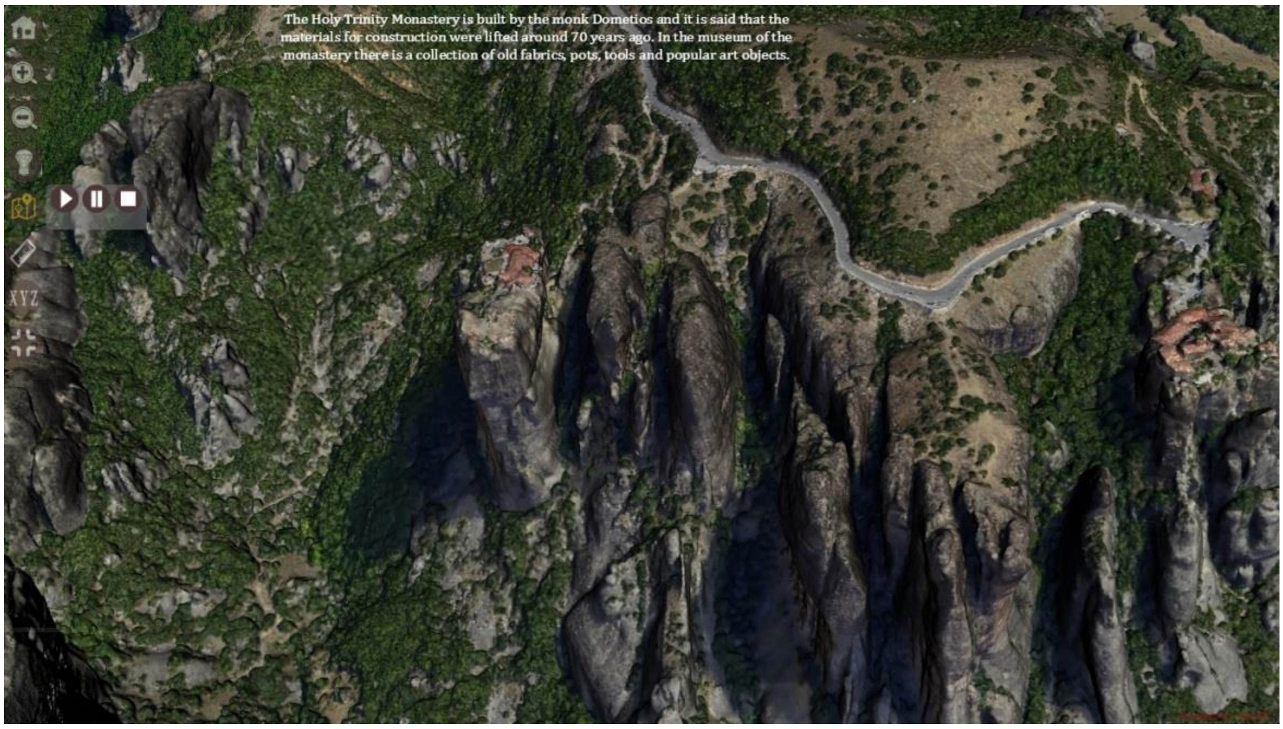
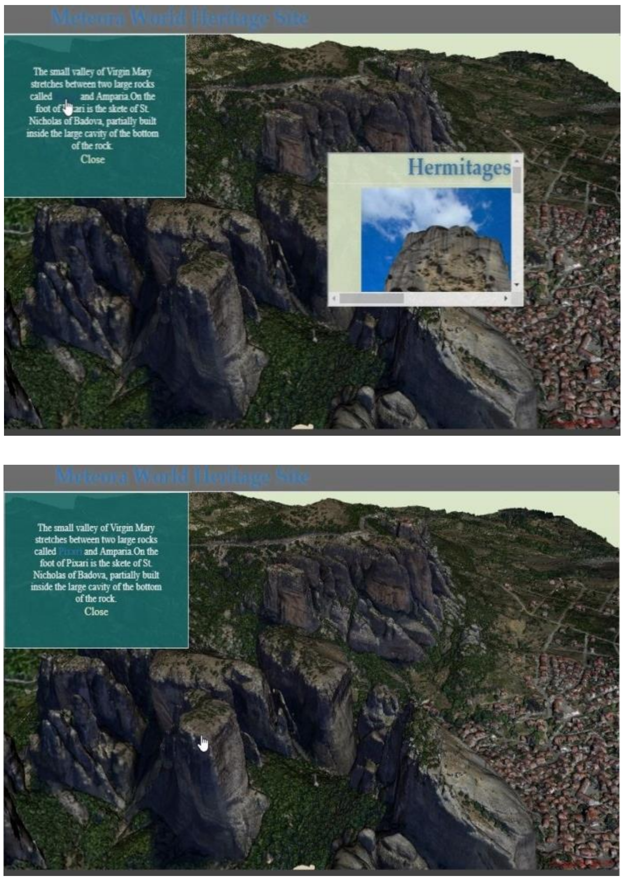

The Holy Trinity Monastery is built by the monk Dometios and it is said that the materials for construction were lifted around 70 years ago. In the museum of the monastery there is a collection of old fabrics, pots, tools and popular art objects.

Figure 4. During the guided tour, a brief description is provided when camera is framing the "Holy Trinity" Monastery.

Frames are used to preview the redirecting page when the mouse is placed over the link (Figure 5). For our case study, the camera frames a specific part of the vast surface to facilitate the user's orientation.

The tool's configuration relies on "hover" event and simple transparent geometry, placed on the desired position over the HTML canvas. A dialog box is opened when the cursor is over the transparent box. The popup is by default hidden and it closes whenever a click is triggered on the box. When the *hover* effect fires, the geometry's ref property is set to the popup dialog box's id, which presumably is different from the transparent box; when the *fadeIn* event fires, the hidden pop up is made visible. A close option is attached to the dialog to cause a fadeout() method; an animation of the pop up's opacity from fully opaque to fully transparent is executed. The pop-up window is actually a modal dialog that, upon activation, prevents interaction with the document until it is closed. Practically, it mimics the annotation structure; an additional set of information is tied on top of the model.



The small valley of Virgin Mary stretches between two large rocks called Pixari and Amparia On the foot of Pixari is the skete of St. Nicholas of Badova, partially built inside the large cavity of the bottom of the rock.
Close

Figure 5. Snapshots during the investigation of the 3D scene of Meteora; in the dialog box, the word Pixari in the description is a link to the 3D visualization of Pixari model and a frame is used for preview.

5. EVALUATION

The visualization of large and high resolution models on browser requires enough network bandwidth, hardware performance and fast data transmission in order to be optimal. The speed of 5 Mbit/s provided by common Internet access technologies is sufficient for 3D streaming. With WebGL, GPU resources are accessed directly while transferring and rendering the 3D data. Therefore, rendering latency is the crucial parameter for the visual result and its evaluation is needed.

The framework comprises two viewers for 3D display: 3DHOP and Three.js. Their main difference lies on the way 3D models are rendered. A tangible feedback about their rendering procedures could be obtained by a performance analysis. The loading time of the main page of our framework was measured on two periods; with only the 3DHOP's viewer and after enabling both viewers.

Analysis was carried out on a conventional laptop with Intel Dual Core i5-7200 CPU at 2.50 GHz, 8 GB of RAM and AMD Radeon™ 520 Graphics (2 GB DDR3 dedicated), over Windows 10. The bandwidth of the network was 9.6 Mbit/s. The multi-resolution version of the model used in the 3DHOP's scene is the archaeological site of Meteora (34M vertices, 2430Kbytes as binary .PLY file) while the model of Three.js loader is a relatively simpler one (3M vertices, 278Kbytes as binary .PLY file). The results of the testing on the latest version of a number of major browsers (Google Chrome, Mozilla Firefox and Microsoft Edge) are presented on Table 1. 3DHOP's rendering has demonstrated a stunning performance. In approximately 3 seconds the high resolution model is defined adequately and the page is fully loaded. The latency of the stand-alone streaming of Three.js is considerable and proves that transferring a big amount of data in one step is not suitable for very large and detailed 3D geometries. In order to minimize the loading time and network traffic, Three.js viewer will not be enabled by default.

	3DHOP scene	3DHOP scene and Three.js viewer
Google Chrome 71	3.18 sec	26.84 sec
Edge 42	3.12 sec	28.60 sec
Mozilla Firefox 63	2.81 sec	27.03 sec

Table 1. Web rendering statistics for 3DHOP and Three.js using the network panel of the Developer tools.

6. CONCLUSIONS

The project done is an online repository of high resolution 3D models with extended capabilities which can be accessed through a common web page. It ensures a responsive and intuitive experience of the unequipped users and serves the needs of researchers of the Cultural Heritage domain. The sophisticated capability of integration of different types of media and the automatic services could be the basis of interactive applications like online cultural heritage repositories, virtual museums and educational presentations. The framework has also the potential to provide developers with prebuilt functionalities and design patterns. At the same time, extensibility is emphasized, encouraging third-party customization. It should be noted that the framework will be available online in the near future.

ACKNOWLEDGEMENTS

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T1EΔK-02859).

REFERENCES

Agugiaro, G., Remondino, F., Girardi, G., Schwerin, J., Rissetto, H., Amicis, R., 2011. QueryArch3D: Querying and Visualising 3D Models of a Maya Archaeological Site in a Web-Based Interface, In: *Proceedings of XXIII International CIPA Symposium*, pp. 10-17.

Anyuru, A., 2012. Professional WebGL Programming: Developing 3D Graphics for the Web, ISBN: 978-1-119-96886-3.

Behr, J., Eschler, P., Jung, Y., Zőllner, M., 2009. X3DOM – A DOM-based HTML5/ X3D Integration Model. In: *Proceedings of Web3D 2009: The 14th International Conference on Web3D Technology*, http://dx.doi.org/10.1145/1559764.1559784

Brunt, P., 2010. GLGE.org, http://www.glge.org/ (Last accessed on December, 2018).

Cabello, R., 2010. Three.JS, https://threejs.org/ (Last accessed on December, 2018).

Cignoni, P, Ganovelli, F, Gobbetti, E, Marton, F, Ponchio, F, Scopigno, R., 2005. Batched multi triangulation. In: *Proceedings of IEEE Visualization Conference (VIS 05)*, Minneapolis, MN, USA, pp. 207–214.

DeLillo, B., 2010. WebGLU development library for WebGL. In: *Proceedings of International Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH 2010, Los Angeles, USA.

Di Benedetto, M., Ponchio, F., Ganovelli, F., Scopigno, R., 2010. SpiderGL: a JavaScript 3D graphics library for next-generation WWW. In: *Proceedings of 15th International Conference on Web 3D Technology*, pp. 165-174.

Galeazzi, F., 2015. ADS 3D Viewer: an example of open 3D real-time visualization system in archaeology. In: *Proceedings of 80th Annual Meeting of the Society for American Archaeology*, San Francisco, California (tDAR id: 397313).

Google, 2008. O3D, https://code.google.com/archive/p/o3d/ (Last accessed on December, 2018).

Guarnieri, A., Pirotti, F., Vettore, A., 2009. An open source application for interactive exploration of cultural heritage 3d models on the web. In: *Proceedings of 3rd International ISPRS Workshop 3D - ARCH'09 on 3D Virtual Reconstruction and Visualization of Complex Architectures,* Trento, Italy.

Kay, L., 2010. Scene.js, http://scenejs.org/ (Last accessed on December, 2018).

Khronos Group, 2009. WebGL - OpenGL ES 2.0 for the Web,

https://www.khronos.org/webgl/ (Last accessed on December, 2018).

Kim, H.M., Rushmeier, H., Ffrench, J., Passeri, I., Tidmarsch, D., 2014. Hyper3D: 3D Graphics Software for Examining Cultural Artifacts, *ACM Journal of Computing and Cultural Heritage,* Vol. 7(3).

Leoni, C., Callieri, M., Dellepiane, M., Rosselli, Del Turco, R., Donnell, D., 2013. The Dream and the Cross: bringing 3D content in a digital edition. In: *Proceedings of Digital Heritage 2013 International Conference*, pp. 281-288 (Journal on Computing and Cultural Heritage - Special Issue on Best Papers from Digital Heritage 2013).

Ponchio, F., 2008. Multiresolution structures for interactive visualization of very large 3D datasets, Phd Thesis, Clausthal University of Technology.

Ponchio, F., Scopigno, R., 2015. 3DHOP: 3D heritage online presenter. In: *Computers & Graphics*, Vol. 52, pp. 129–141.

Potenziani, M., Callieri, M., Dellepiane, M., Corsini, M., Scully, T., Dobos, J., Sturm, T., Jung, Y., 2015. 3drepo.io:building the next generation Web3D repository with AngularJS and X3DOM. In: *Proceedings of 20th International Conference on 3D Web Technology*, pp. 235-243.

Soler, F., Melero, J.F., Luzon, M.V., 2016. A complete 3D information system for cultural heritage documentation, *Journal of Cultural Heritage*, vol. 23 (January–February 2017), pp. 49-57.

Sons, K., Klein, F., Rubinstein, D., Byelozyorov, S., Slusallek, P., 2010. XML3D: interactive 3D graphics for the web. In: *Proceedings of 15th ACM International Conference on 3D Web Technology*, Los Angeles, USA, pp. 175-184.