

Artworks narrating a story: a modular framework for the integrated presentation of three-dimensional and textual contents

Marco Callieri, Chiara Leoni, Matteo Dellepiane, Roberto Scopigno
Visual Computing Laboratory, ISTI-CNR, Pisa, Italy
e-mail:surname@isti.cnr.it

Abstract

The integrated presentation of a 3D model and a descriptive text which is strongly connected to the geometry (and, more in general, of multiple interconnected datasets of different nature) is still problematic. This is often solved by using plugins and/or resulting in poorly connected exploration tree (where one media is used as a "primary content", and the rest of the data are just used to enrich it).

In this paper, we present a framework for the presentation, on the web platform, of multiple heterogeneous datasets. Thanks to a modular design, and a simple mechanism to index data structures and messages, it is possible to design interconnected viewers, and build a flexible system for this purpose. By explicitly annotating the datasets with locators, which are present in all datasets, it is possible to build a strongly connected graph, and let the user explore it, obtaining a real multi-media exploration, *inside* and *across* the various datasets. The entire system is implemented using HTML5 and JavaScript, making it a portable solution, able to run natively inside browsers, without additional components. We tested the system on two practical, complex examples, both composed of an intricate high-resolution geometry, and rich textual information.

CR Categories: I.3.6 [Computer Graphics]: Methodologies and techniques—Graphics data structures and data types I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—;

Keywords: WebGL, multiresolution, presentation

1 Introduction

Three-dimensional models are nowadays becoming a standard ingredient of the daily information flow that is made available to the public. The means of creation of the models (3D modeling, reconstruction from images, 3D scanning) and their usage in the process of presentation call for different types of implementation, interaction and quality.

In the context of Cultural Heritage, the use of highly detailed 3D models is often necessary to be able to fully represent and present the peculiarities of an artifact. This need of detail is due to the fact that almost any decorated Cultural Heritage object (i.e. carved surfaces, basreliefs, façades) exhibits both a *macrostructure* and a *microstructure*. The macrostructure is basically the global shape

of the object, generally determined by its function, and its context. The microstructure is represented by the nature, shape and significance of its decorations and their inter-relations. Being able to show both levels at the same time is essential for the comprehension of the artifact. We can bring, as an example, the famous Trajan's Column (http://en.wikipedia.org/wiki/Trajan's_Column). Analyzing such an artifact is often like following a story (narrated by the elements of its microstructure) which is winding across the global structure of the object, moving from a decoration to another one. Different stories and concepts can be found by following different paths; we may say that a 3D model representing an object of this class does 'tell a story'.

Presenting one of these objects is not an easy task; focusing in the framework of teaching, a user (a student or scholar) should be able to explore the microstructure following a path of its own design (freely roaming on the 3D surface). At the same time, he should have access to a solid textual reference for all the elements of the decoration (free textual exploration). Finally, there should be always a link between the textual and the 3D data (synchronization between 3D and text view). This concept of free exploration of a dataset and synchronization across all its elements could be further extended to other media (a set of photos, videos, technical drawings and surveys). Additionally, it would be extremely useful, for a teacher preparing a lesson or for a student preparing an essay, to build a specific, annotated, exploration path of the available datasets.

This paper presents a modular, extendible 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 refer to a set of reference points that link to all the modules of the presenter. The system is entirely based on Javascript, HTML5 and, in particular, WebGL; this enables the system to work without external components and plugins. The main contributions of the work are:

- The independence of the content (3D, text) with respect to the visualization infrastructure (HTML5 + Javascript). The module manager generates the webpage by interpreting a small set of xml files, which can be easily generated and modified even without a deep knowledge of web development.
- The evolution of usual links between the different types of data to the concept of spot, an entity which is stored independently of the types of data shown in the presenter. The spots interconnect all the modules in the presenter, building over the concept of mono-directional link.
- The definition and easy creation of 'lectures', which are pre-defined visits of the 3D model which follow the defined spots in order to present specific paths, or tell stories following not only the text, but also the spatial path around and inside the object.

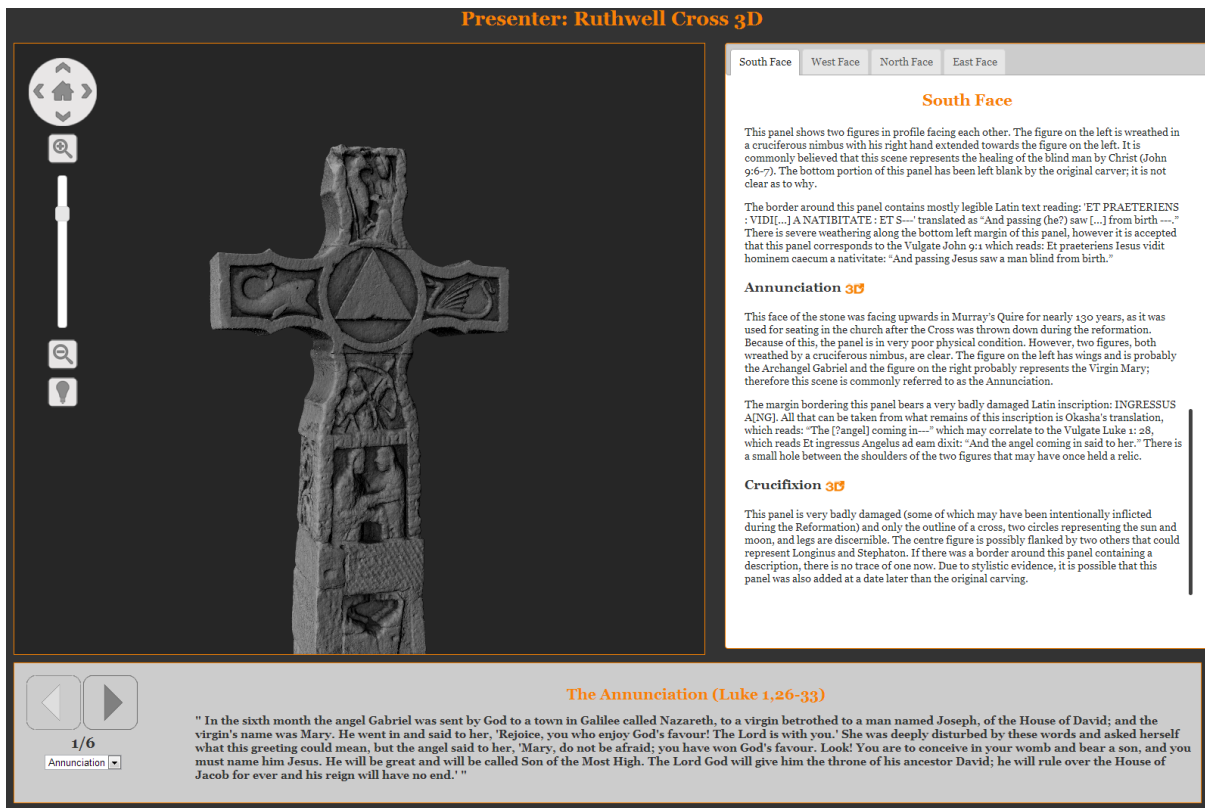


Figure 1: A screenshot of the proposed system, presenting the Ruthwell Cross.

2 Related work

2.1 Integrating different media in a common environment

The use of multimedia to integrate and enhance the presentation of complex data was studied and conceptualized since the advent of the advanced ways to visualized images and videos [Faraday and Sutcliffe 1997]. Also the potential of the integration of 3D data with other types of information was clear right from the birth of Internet. Unfortunately, their use was delayed by the lack of a standard format for data and visualization (see next Subsection). Nevertheless, virtual environments enriched with textual information have been thoroughly studied in all their aspects, including not only navigation, but also the disposition and readability of the text [Sonnet et al. 2005; Jankowski et al. 2010], and the usability of interaction paradigms [Bowman et al. 2003; Polys et al. 2011].

All the works above mentioned described a inherently three-dimensional environment, where textual information was attached to the models in a spatially coherent way. This approach did not handle a data structure where different types of media could be interconnected in a multi-directional fashion. The basics of this structure were theorized by Halasz and Schwartz [Halasz and Schwartz 1994], and the principles of approaches based on the integration of different media in a single application were described even before [Yankelovich et al. 1988].

More recently, two works showed approaches to combine 3D models and textual information in an integrated fashion. The first

one [Götzelmann et al. 2007] presented a system to link textual information and 3D models. The links on the text bring to the display of pre-defined points of view on a 3D model, while the arbitrary definition of a point of view activates a mechanism that proposes the more adapt textual information. This method shares several aspects with our one, but it was implemented as a stand-alone application. Another recent work [Jankowski and Decker 2012] explored the possibility of the integration of text and 3D models on the web, but it proposed two arbitrary visualization modes: one is a typical hypertext, while the other one creates a virtual 3D environment, where text is attached to 3D hotspots.

2.2 3D graphics on the Web platform

The Internet have been a source of different data type right from its birth. Text, images and videos are native ingredients of a web page. With the advent of graphics acceleration hardware, also 3D content gained an important role. In the beginning, the visualization of 3D models in Web pages was devoted to embedded software components, such as Java applets or ActiveX controls [ACT]. The lack of standardization for this new content type led to the development of several proprietary plug-ins and extensions for Web browsers. The need to install external components prevented from a wide use of these different solutions.

Virtual Reality Modeling Language (VRML) [Raggett 1995] in 1995 and X3D [Don Brutzmann 2007] in 2007 were action aimed at finding a standardization. However, even though they have been well-accepted by the community, the 3D scene visualization was still delegated to external software components.

The introduction of the WebGL standard [Khronos Group 2009c], promoted by the Khronos Group [Khronos Group 2009a] in 2009, was a fundamental change. The WebGL API is a one-to-one mapping of the OpenGL|ES 2.0 specifications [Khronos Group 2009b] in JavaScript. Hence, modern Web browsers are able to *natively* access the 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 non-expert users. They differ from each other by the programming paradigm they use, ranging from scene-graph-based interfaces, like Scene.js [Kay 2009] and GLGE [Brunt 2010], to procedural paradigms, like SpiderGL [Di Benedetto et al. 2010] and WebGLU [DeLillo 2009]. The X3DOM [Behr et al. 2009] model directly integrated X3D format with WebGL. X3DOM is currently one of the most robust solutions for the integration of 3D data in websites.

WebGL has been already used to implement complex rendering and interaction techniques in several fields, like bio-informatics [Callieri et al. 2010], volumetric data visualization [Congote et al. 2011], interior design [Priyopradono and Perdana 2013]

3 Reading a 3D model: the modular framework

The aim of this paper is to present a simple, extensible framework for the remote visualization of the different datasets (i.e. text, 3D model, images, videos) which describe a Cultural Heritage artwork. This type of object is often represented with multiple, heterogeneous, highly interconnected datasets.

Our vision is therefore that a presentation system should be organized in a modular and interconnected manner, giving to the user all the features needed to structure the navigation and visual presentation of the interconnected and interrelated media. Therefore, in our approach the visualization page is composed by a series of *viewer modules*, each one tailored to a specific type of data, allowing free navigation. All the viewers communicate through a *manager module*, able on user request to synchronize all viewers. Additionally, another *lecture module* provides pre-defined exploration paths for all the datasets. A user would then be able to start browsing a dataset (e.g. the 3D model, moving spatially across the artifact surface), then synchronize all viewers, and continue its exploration in another media (e.g. the structured text, following the story narrated by the decorations). Alternatively, the user may want to start following a predefined path of exploration (using the "lectures" functionality) and then, at any time, wander off the path, continuing a free exploration in a media of his choice. This will make a *true* multi-media exploration, instead of having a "main" media (generally, the text or the 3D model) acting as the exploration tree, and multiple media objects to act as leafs of this tree.

In Figure 1, it is possible to see the two main modules already implemented: the 3D model viewer (left side) and the structured text viewer (right side); plus, the lecture module (below).

One of the most important features is that the data visualization of the webpage is *completely parametric* w.r.t. the content. Each visualization module is independent: it is configured using an XML file, it is able to read a specific kind of dataset, and it provides a way to visualize it. The user may freely browse the dataset or explicitly ask for a synchronization of all viewers, using an interface element/action specific to the kind of dataset (click on a link in the text, double-click on a portion of the 3D model, and so on). A simple mechanism of messages is used by the viewers to communicate with the manager module, to ask for a general synchronization. The manager module, also configured through an XML file, is responsible for the interaction between the viewers. This module contains a "table" with all the defined synchronization points, called *spots*. Each spot is basically a semantic element which defines a "location" in each dataset. When a viewer asks for a synchroniza-

tion, it indicates one of these spots. All the viewers, receiving the "goto(spot)" command from the manager, change their view over the dataset by focusing on the same spot. These navigation spots are built upon the concept of single-direction links, by providing a way to effectively merge a series of datasets in a strongly connected graph, resulting in a seamless exploration.

This free in- and intra-dataset exploration may also be integrated with another element, the *lecture module*, used to display predefined exploration paths. Using a simple slides-based presentation, the module guides the user across all the datasets, jumping from a spot to another.

This basic scheme is quite simple to implement using JavaScript, it is easy to configure and edit by customizing XML files but, at the same time, it is powerful and extensible. The independence of the viewers and the simple message mechanism enable to introduce new types of data, by simply introducing a new viewer.

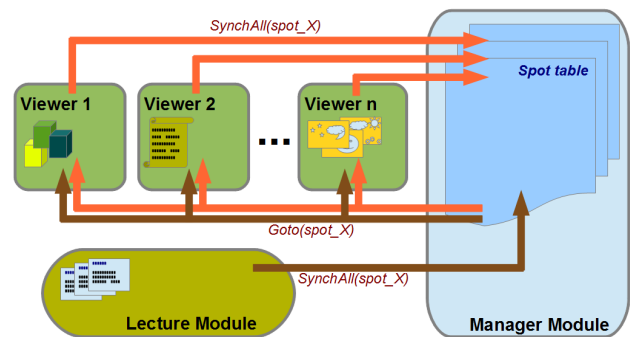


Figure 2: A scheme of the system: Viewer modules communicate with the Manager module by calling for a synchronization, the Manager receive the message and broadcast a command to focus on the given spot. The same message path is used by the Lecture module; when a new slide is presented, a signal is sent to the Manager, and all the Viewers are synchronized.

A scheme of the interaction of the modules is shown in Figure 2: the Manager module is simply a message dispatcher, that may receive the command to synchronize all viewers by one of the Viewers (when the user selects a spot) or from the Lecture module (when a new slide is presented). The Manager module, upon receiving the SynchAll command, broadcasts to all Viewers the signal to focus on the requested spot.

The html page is almost empty: the various modules are arranged in the page using standard html formatting, the XML configuration files are specified at the start of the script area, and some startup code is used to initialize the modules. Changing the subject of the visualization is just a matter of changing the configuration files.

All the configuration and some dataset definition (like the structured text) is done using XML files. This is a completely arbitrary choice: JSON encoding, XML or even a custom data format are equivalent. XML was chosen because the non-computer science people involved in the project (mainly Cultural Heritage operators) were more familiar with this kind of encoding, and already had all the proper tools to edit this data format (editors, viewers, exporters from their databases).

In the following sections, we will give a short description of the modules and XML configuration elements.

3.1 3D Viewer module

This module is used to show the three-dimensional model of the object of interest. Since it is based on WebGL and SpiderGL library, it runs natively in the browser, without requiring external plugins or components.

Thanks to the higher lever wrappers to WebGL and the computer graphics helper provided by SpiderGL [Di Benedetto et al. 2010], it was possible to bring advanced CG techniques, like multiresolution, on the web platform. The rendering of the 3D model uses the Batched Multi Triangulation approach [Cignoni et al. 2005]. This multi-triangulation method re-organizes the different detail level of the models using triangle patches as the basic element. The rendering strategy and the data structure are well suited for remote visualization: the server will provide the need 3D resolution following a view-dependent metric. Hence, if the model is visualized from afar, a low resolution model is shown. If the user focuses on a detail, only the high resolution geometry of that portion will be downloaded. This visualization strategy provides a remote visualization of huge 3D models (up to hundreds million triangles, as shown in the Section 4) with virtually no visual artifacts and without the need of advanced hardware on the client side.

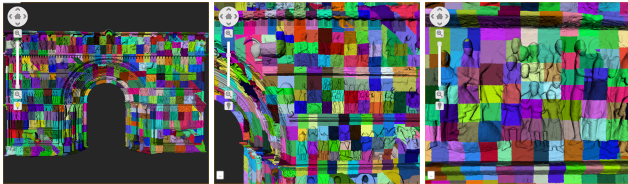


Figure 3: The multiresolution engine underlying the 3D viewer: while approaching to the object, smaller higher-resolution patches are fetched from the network, and rendered.

In Figure 3, it is possible to see the underlying multiresolution structure: the whole geometry is divided in chunks, which are fetched from the network using asynchronous requests. Depending on the viewer distance, for each part of the object an "optimal" resolution is chosen, and the needed patch, if not already loaded, is requested. While approaching the objects, smaller and higher-resolution patches will appear. The main advantages of this multiresolution scheme is that the viewer will have something to show as soon as it starts (a very low resolution model), and the quality of the model will be refined as the exploration continues, when higher resolution patches are loaded. Moreover, only the parts that are examined more closely will be transferred over the network.

The viewer employs a simple shading for rendering, more advanced rendering methods may surely be introduced in the future, to cope with specific needs of datasets, thanks to the shader-based structure of WebGL. The user may change light direction, a very useful tool to better visualize carvings (as shown in Figure 11).

The viewer is basically an empty canvas element, where WebGL accounts for the rendering. The navigation interface on the left is just a series of html elements, connected to functions of the viewer. The navigation of the 3D object has been kept simple on purpose: it uses only the left mouse button and motion, in order to be usable on a touchscreen.

The free navigation over the 3D model is based on a simple trackball. It is obvious that different objects do require a different exploration method; for this reason, the type of trackball is one of the parameters of the scene. In the Cross example, a cylindric trackball was used, while a pan-tilt trackball was used for the Portalada (see Section 4). Additional trackballs may be introduced

to cope with specific characteristics of the object to be navigated. This does require a bit of skill in analysing the model, but starting from the existing trackballs and thanks to the simple interaction between the viewer and the trackball, the task is not difficult.

While hovering with the mouse over an "active" area of the object, semi-transparent panels appear, to show the user that an area is covered by a spot, and a synchronization may be asked. These panels have been created manually inside an external tool (i.e. using Blender) and exported as a list of points. See section 3.5 on data authoring for more details. Double click on one of such panels triggers the synchronization of all the viewers.

When the module receives the message to synchronize to a given spot, the view position of that specific spot is retrieved from the dataset and the rendering is modified accordingly. The viewpoint is animated smoothly from the current to the desired position; the animation when going from one spot to another is an important feature, since it helps the user not to get lost.

Figure 4 shows the format of the XML describing the 3D scene for the Cross example in Figure 1.

```
1 <scene scene_title="Ruthwell Cross">
2   <background>
3     <color>[ 0.15, 0.15, 0.15, 1.0 ]</color>
4   </background>
5
6   <meshes>
7     <mesh id="crossmesh" url="Cross_120M.nxs"></mesh>
8   </meshes>
9
10  <modelInstances>
11    <Instance id="cross" mesh="crossmesh"></Instance>
12  </modelInstances>
13
14  <trackball type="CYLINDRIC">
15    <trackCenter>[0.0, 0.0, 0.0]</trackCenter>
16    <minMaxDist>[515, 6500]</minMaxDist>
17    <minMaxPan>[-5050, -110]</minMaxPan>
18    <startRotation>0.0</startRotation>
19    <startDistance>6500.0</startDistance>
20    <startPan>-2500.0</startPan>
21  </trackball>
22 </scene>
23
24 <spot_table>
25   <spot spot_id="sf_archer" >
26     <trackPosition>[-0.40185, -4433.7624, 515.0]</trackPosition>
27     <points>
28       [
29         127.590874, 4625.775879, 168.106567,
30         -139.849335, 4625.775879, 168.106567,
31         155.227036, 4316.472168, 168.106567,
32         -184.693405, 4316.472168, 168.106567
33       ]
34     </points>
35     <normal>[ 0.0, 0.0, 1.0 ]</normal>
36   </spot>
37
38   <spot spot_id="sf_visitation" >
39     <trackPosition>[6.0591, -3953.4404, 1015]</trackPosition>
40     <points>
41       [
42         155.572922, 4314.296875, 168.106567,
```

Figure 4: An example of the XML configuration file describing the 3D scene.

Beside some basic description of the scene (i.e. background color), two elements have to be defined: the 3D model, and the trackball. The 3D model is defined as single instance of a mesh, which is read from a file containing the multiresolution structure. The trackball, used for the navigation, is configured specifying its type (in this case, cylindric), and providing parameters describing the limits of user movement (chosen accordingly to the size of the current 3D model), and the starting position. Additionally, the XML configuration file contains the "spatial" information for all the spots, used by the viewer to focus to a given spot (upon

receiving the synchronization command), and to render and let the user pick the area of the spots. This is, for each spot, a set of trackball parameters used to frame the interested area, and a series of points, used to build a simple polygon to be rendered (and picked). This information could also reside in the main spot table inside the Manager module but, given its spatial nature and the fact that it is strongly correlated to the 3D model, it was decided to store it inside this module.

3.2 Structured Text module

This module is used to present a structured text to the user. The text, describing the object, is structured in the sense that is divided in sections (rendered as tabs) and subsections (rendered as titled paragraphs), according to some logical or spatial properties. For example, in the case of the Cross, the text describing the carved panels was divided in four sections, one for each side; and, for each side, there is a subsection describing each panel.

Each tab is connected to a spot, and changing tab invokes a synchronization. Similarly, the paragraph titles (each one connected to a spot) act as links. When the module receives a synchronizing command, it changes the tab and scrolls the text until the paragraph describing the selected spot is displayed. The paragraph is also shortly highlighted.

Since the text is almost static, i.e. the functionalities to display the text, interact with tabs and scrollbars, and select a spot for the synchronization can be achieved easily enough with standard html elements (plus some help from JQueryUI). However, to maintain the independence between the visualization page and its content, the html code is not hard-coded into the page, but generated when the page is loaded. The structured annotated text is read from the XML configuration file into a structure in memory, transformed in html and injected into an empty DIV.

Figure 5 shows a portion of the XML file describing the configuration for the structured text module, as it is used in the example in Figure 1.

```

1 <tabs>
2 <tab spot="south_face" title="South Face">
3
4 <panel spot="sf_top" title="Top Piece">
5 <p>John the Evangelist. This panel depicts a man hold a bird and is commonly accepted to be John
6 the Evangelist with his emblem, the eagle. The inscription in the border is mostly illegible due
7 mainly to weather, although some damage may have been incurred during the reformation, when the Cross
8 was torn down. The right corner of the panel is significantly damaged, which may have been caused by
9 falling to the ground. However, there are traces of the Latin letters: IN [... ..] UM, generally
10 expanded to in principio erat verbum, which are the first words of the Gospel of St. John.</p>
11 <p>It is thought now, that when Rev. Henry Duncan rebuilt the Cross in 1823, that he placed this piece
12 on backwards, that John and his eagle were meant to balance out the North side and be above Matthew
13 and his angel. The image on the top of the South face was intended to be the solitary eagle, and take
14 its place above the Archer, the panel below the modern transept. The panel, along with the Visitation
15 panel, was found by Rev. Duncan buried in the ground.</p>
16 </panel>
17
18 <panel spot="sf_archer" title="The Archer">
19 <p>This panel makes up the lower arm of the cross-piece. It depicts a man in profile drawing his bow,
20 and is referred to as "The Archer." Beneath the figure's arm is a square object on a strap, and is
21 most probably a quiver. There is some evidence that there was once an inscription bordering this panel,
22 however, because of severe weathering, it is completely illegible.
23 If the top piece were faced the right way around, the archer would be paired with the solitary eagle
24 found on the North face.</p>
25 </panel>
26
27 <panel spot="sf_visitation" title="The Visitation">
28 <p>This panel was found buried in the ground by Rev. Duncan. This panel depicts two figures in profile,
29 facing each other and in an embrace. There is some confusion as to who these two figures are (Mary and
30 Martha or Mary and Elizabeth), which stems mainly from the translation of the inscription bordering this
31 image, as it is extremely weathered.</p>
32 <p>The text is a mixture of Latin and Runic text. The Latin can roughly be read as M[...R]JA . M[...]ER
33 . DOMINIANE. There are a number of translations offered, and some include "Mary, Mother of the Lord,"
34 "Martha and Mary, Mother of the Lord" or "Martha and Mary, meritorious ladies." The Runic text is read
35 as --EFAVEIC-- and DEBISSEY[...], but the meanings are uncertain. The blank concrete squares were added
36 by Duncan to fill in the missing pieces. Indeed, there is a chance that the feet in this panel do not
37 actually belong here.</p>
38 </panel>

```

Figure 5: An section of the XML configuration file containing the structured text.

The two-level structure of the text is encoded in a nested XML structure; the description of each panels is inside a *panel* element (each one specifying a title and a spot), an the panels are grouped

into *tab* elements (again, specifying a title and a spot). Since the spot ID for each element is already specified in each XML tag, all the need information used by the viewer to display spots for picking and to scroll the text when a synchronization command is received. At the moment, the formatting of the text is quite limited, and it is only supporting a basic subdivision in paragraphs. We are currently working on a better annotation system, to be able to render in a more structured way text elements like transcriptions, formatted poetry, and special symbols.

3.3 Manager module

While this is the core module, used by the viewers for the synchronization, it is the simplest entity in the system. The manager contains the table of spots and acts as a message dispatcher. When one of the viewers asks for a synchronization, the manager sends a message to all the viewers containing the unique ID of the selected spot. The manager is also used by the lecture module, when the predefined path is presented to the user.

Figure 6 shows some of the spots defined in the XML configuration file for the manager, used in the example shown in Figure 1.

```

1 <spotstable spotstable_title="The Ruthwell Cross">
2 <!-- GENERAL -->
3
4 <spot spot_id="start" >
5 <tooltip>Default full cross view</tooltip>
6 <color>[0.5, 0.5, 0.5, 1.0]</color>
7 </spot>
8
9 <spot spot_id="south_face" >
10 <tooltip>Cross face facing the church altar</tooltip>
11 <color>[0.8, 0.5, 0.5, 1.0]</color>
12 </spot>
13
14 [... ..]
15
16 <spot spot_id="east_face" >
17 <name>East Face</name>
18 <tooltip>Cross face facing the right church wing</tooltip>
19 <color>[0.5, 0.5, 0.8, 1.0]</color>
20 </spot>
21
22 <!-- SOUTH SIDE -->
23
24 <spot spot_id="sf_top" >
25 <name>South Top Piece</name>
26 <tooltip>John the Evangelist</tooltip>
27 <color>[0.2, 0.2, 0.2, 1.0]</color>
28 </spot>
29
30 <spot spot_id="sf_archer" >
31 <name>The Archer</name>
32 <tooltip>The Archer drawing its Bow</tooltip>
33 <color>[0.2, 0.2, 0.6, 1.0]</color>
34 </spot>

```

Figure 6: Some spots, as specified in the XML configuration file used by the Manager module.

The parameters of the spot are just, at the moment, the unique ID and an extended name. Additionally, some other general purpose fields are used to specify some properties, like the tooltip and a color value, that may be used by all the viewers when rendering the spot; in this way, these values will appear uniform throughout the presenter. The actual location of the spots inside each dataset (for rendering and picking), is stored in each viewer, in the annotated dataset.

3.4 Lecture module

This module is used to present predefined exploration paths to the user. The paths are built, in form of "lectures", using the available spots. A lecture is a series of elements called *slides*, composed by a title and a text, which are connected to one of the spots. When the slide text is presented, the manager is triggered to synchronize all the viewers to the correct spot.

The XML configuration of the module, shown in Figure 7, is quite simple. Additional elements, like images, or a more structured formatting may be introduced in future versions.

```
1 <|lecture lecture_title="The Cross and the Gospels">
2
3 <|slide slide_id="Annunciation" slide_title="The Annunciation (Luke 1,26-39)" slide_spot="sf_annunciation">
4 <p> " In the sixth month the angel Gabriel was sent by God to a town in Galilee called Nazareth, to a virgin
5 betrothed to a man named Joseph, of the House of David; and the virgin's name was Mary. He went in and said
6 to her, 'Rejoice, you who enjoy God's favour! The Lord is with you.' She was deeply disturbed by these words
7 and asked herself what this greeting could mean, but the angel said to her, 'Mary, do not be afraid; you have
8 won God's favour. Look! You are to conceive in your womb and bear a son, and you must name him Jesus.
9 He will be great and will be called Son of the Most High. The Lord God will give him the throne of his
10 ancestor David; he will rule over the House of Jacob for ever and his reign will have no end.'" </p>
11 </slide>
12
13 <|slide slide_id="Visitation" slide_title="The Visitation (Luke 1,39-45)" slide_spot="sf_visitation">
14 <p> "Mary set out at that time and went as quickly as she could into the hill country to a town in Judah.
15 She went into Zechariah's house and greeted Elizabeth. Now it happened that as soon as Elizabeth heard Mary's
16 greeting, the child leapt in her womb and Elizabeth was filled with the Holy Spirit. She gave a loud cry and
17 said, 'Of all women you are the most blessed, and blessed is the fruit of your womb. Why should I be honoured
18 with a visit from the mother of my Lord? Look, the moment your greeting reached my ears, the child in my womb
19 leapt for joy. Yes, blessed is she who believed that the promise made her by the Lord would be fulfilled.'" </p>
20 </slide>
21
22 <|slide slide_id="Egypt" slide_title="The flight into Egypt (Matthew 2,13-15)" slide_spot="sf_egypt">
23 <p> " After they had left, suddenly the angel of the Lord appeared to Joseph in a dream and said, 'Get up,
24 take the child and his mother with you, and escape into Egypt, and stay there until I tell you, because
25 Herod intends to search for the child and do away with him.' So Joseph got up and, taking the child and
26 his mother with him, left that night for Egypt, where he stayed until Herod was dead. This was to fulfil
27 what the Lord had spoken through the prophet: I called my son out of Egypt. " </p>
28 </slide>
```

Figure 7: An example of a lecture xml descriptor

Multiple lectures may be defined in the configuration file. The module renders in a DIV a simple interface (visible in Figure 1, on bottom), letting the user choose the lecture and follow the slides.

3.5 Dataset Authoring

Since our system is basically a presenter, it is important to also discuss how the content is created. Each viewer in the system is using an "annotated" dataset; annotated in the sense that some points/locations/areas of the dataset have been marked with a certain spot ID. The authoring is then, basically, the generation of a list of spots, and the marking of datasets.

In the examples presented in this work, all data were manually prepared. The structured text was generated by isolating parts from existing texts, or written explicitly for this tests by experts. Starting from the text, already partitioned in sections, the annotation was quite simple, only requiring the encapsulation of the text portions inside the appropriate XML tags. The view parameters in the 3D viewers, used to set the focus of the viewer to the various spots were manually defined using the viewer itself. Just by moving to the desired position and pressing a key, it was possible to save the view parameters, and add them to the dataset. The definition of the semi-transparent panels was a bit more complicated; the panels were modeled inside Blender, and exported as a series of points. While this procedure is not very long (one hour to define all the panels for the cross), it is nevertheless complex, and it would become much more accessible with a simple editor integrated into the viewer.

While this completely manual method was effective, and not too cumbersome, the only way to make the system really usable, is the introduction of an authoring system. Given the needed data, it would be possible to implement it completely as a web-based application; most of the data can be entered in simple forms, or edited/configured in modified versions of the viewers. This

would be especially true for the authoring of the 3D elements: the 3D viewer could be modified to include a simple interface for the definition of views and the creation of the clickable panels. This "visual" interface could be really helpful for users without experience in data encoding, XML and web programming.

Additionally, a different option would be just supply a series of XML templates, to be filled by the user with information, using an XML editor of its choice. A series of XML schemas would then be used to validate the prepared files, before being loaded by the system. This is a more manual approach, but could be faster for users (like web designers and copywriters) which are used to manually edit configuration/content files.

Finally, since all the configuration and dataset files are XML, it is also possible to write simple exporters from already existing tools and databases, especially for those kind of data like the structured text or the lecture. In this way, it would be possible to convert pre-existing material to be used in this system. This final option would be useful when a bulk of data which fits nicely in this presentation paradigm was already available, and a manual conversion would be too cumbersome.

4 Examples and discussion

The modular framework presented in this work was tested on two real cases. The examples had a high resolution 3D model available, and shared some common features, like the highly decorated surfaces and the symbolic and narrative meaning of each sub-portion, which could be linked to others following different conceptual paths.

Nevertheless, the two objects were different in shape, in number and in arrangement of the narrative elements, so that their navigation needed to be managed in a different way.

The first example was the Ruthwell Cross. The Ruthwell Cross is a stone Anglo-Saxon cross probably dating from the 8th century. It is both the most famous and elaborate Anglo-Saxon monumental sculpture, and possibly the oldest surviving "text" of English poetry, predating any manuscript containing Old English poetry. On two sides, it shows (in runic alphabet), portion of the poem of *The Dream of the Rood*.

The Ruthwell Cross, like most of the Anglo-Saxon and Irish crosses of the same period, presents narrative carvings, mainly arranged in panels: the north and south sides contain figurative relief carvings, while the west and east sides are decorated with vine tracery. Latin and Runic text inscriptions are present.

This important artifact, and in particular its relationship with the poem is the focus of the the Visionary Cross Project (<http://www.visionarycross.org/>). This interdisciplinary initiative is very interested in the study of this 3D-textual relationship, and the generation of the 3D model of the Cross was also an effort to study the poem text in its three-dimensional space. The Cross was the object that started the development of the presented system: the original idea was to display the 3D model of the Cross, plus a descriptive text, strongly linked to the digital model, to be used as a school text for the scholars and experts of Anglo-Saxon studies. The viewer modules and the manager would represent a sort of online textbook, to be used as a support for teaching and studying. The lecture manager was introduced both for teachers (to prepare, as the name says, a lecture using the material contained in the online textbook) and for students (to prepare essays). The plan was to then go a deeper level, and establish the linkage between text and 3D not at the level of panels, but at the level of the carved runes. During the development, however, it appeared clear that the system could be made into a more flexible tool, configurable and usable to

present an entire class of object with similar characteristics. The presenter content was obtained using a high resolution 3D scanned model of the Cross, and the consolidated description of the elements of the Cross. The model shown in the screenshots is a 120 Million triangle mesh, with ambient occlusion baked on. Based on this, the list of spots linking textual content to view position (see Figure 8) was easily created, and a lecture following the Gospel stories in a chronological way was added.

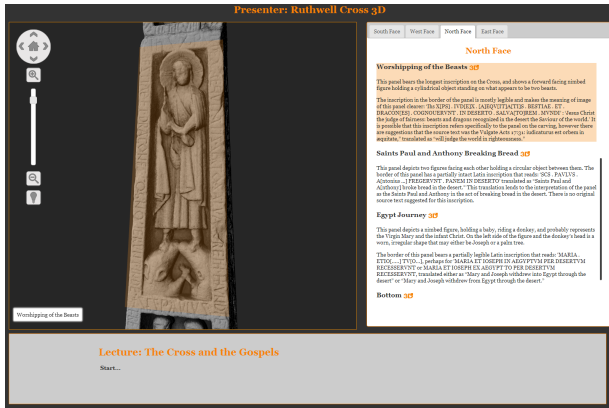


Figure 8: Picking a spot

The 3D navigation is structured in a cylindrical fashion: right and left mouse drags produce a rotation around the object, while the up and down drags shift vertically the cross along its axis. The choice of a constrained navigation, while limiting the possible viewpoints, greatly simplifies the visualization of an object by providing an interaction which is tailored to its shape.

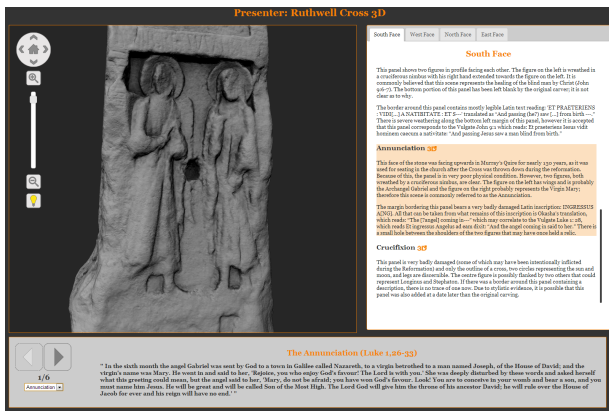


Figure 9: Following the Lecture

Figure 9 shows the visualization when the lecture is followed. In this case, both the 3D and the text modules are updated to show the chosen spot, so that the user can take advantage of the 3D view, the (highlighted) basic text, and the additional information in the lecture model (bottom part of the screen).

The second test case was the Portalada of Ripoll. The Portalada is the decorated arc of the entrance of Santa Maria monastery, located in the town of Ripoll. The portal dates around the middle of the twelfth century, and it is one of the most monumental examples in terms of the sculptural decoration, in the field of

Catalan Romanesque school. It is shaped as a triumphal arch, and it has a decorative profusion of Old Testament scenes carved by anonymous sculptors, following tales of the Bible. Also in this case, a highly detailed 3D model was available [Callieri et al. 2011]; the model used in the test was a 170 Million triangles mesh, with photographic color mapped per-vertex.

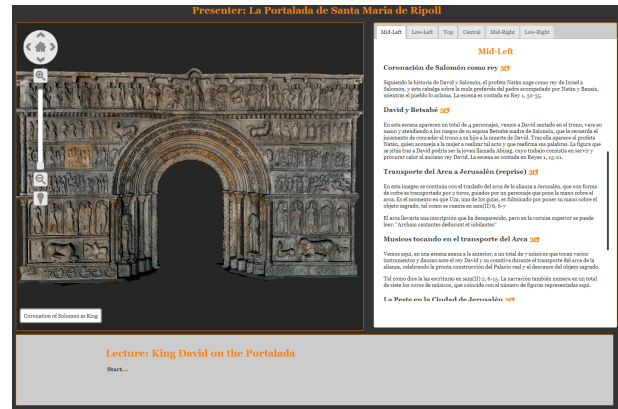


Figure 10: The presenter configured for the Portalada de Ripoll, the same layout of the cross, but with the Portalada content.

While searching for material, we discovered a Website (www.romanicoatalan.com/Ripolles/Ripoll/ripoll03.htm), providing a detailed description of the portal, which was already in a suitable format for the system, because it divided the monument in zones, and described each carved scene in detail. Moreover, additional information about the "stories" of the Bible followed paths through the portions of the portal. While extremely detailed and informative, the Webpage was structured in a classic fashion, resulting in a series of very long pages, so that a non-linear, spatial exploration was nearly impossible. Hence, we decided to try transferring the information of the page on our presenter. The same high-level and low-level (respectively the tabs in the text module and the selected panel, both visible in Figure 10) subdivision was chosen, and the textual information was assigned to each spot of the geometry and to the elements of the lectures.

The new presentation proved to be successful, because the structured organization of the text was enhanced by the availability of a high quality 3D model, which strengthened the description of the portions, providing new ways to explore the surface. For example, Figure 11 shows how a simple change of light direction can greatly enhance different features of the object of interest.



Figure 11: A portion of the Portalada de Ripoll under different light conditions (left, frontal and right illumination direction)

Due to the different shape of the portal, which was more similar to a classic basrelief, the constrained navigation was modeled following a pan-tilt strategy, where the user could freely move in front of the object and tilt its view direction. Due to the flexible structure of the framework, the change of navigation fashion was

obtained only by indicating a different option in the trackball description of the XML scene.

The creation of the textual content for this example took two hours of work using a common text editor, showing that configuring the framework for a new object is fully possible, thanks to the separation of the content w.r.t. the visualization code.

The examples above show how it is possible to easily create a presenter for complex 3D objects with structured textual information. The framework can be easily adapted to any type of *tutto-tondo*, planar or cylindrical objects, but it could also be used for indoor or outdoor spaces like rooms or squares (introducing other navigation paradigms). The direct application to more structured objects, like archeological or architectural sites, could be more difficult due to the issues related to navigation and exploration of multiple environments. Nevertheless, a solution to this could be to create a set of presenters, one for each room or peculiar place, possibly linked through a higher level structure acting as an index.

5 Future additions

The work presented in this paper is still ongoing and, exploiting the modularity of the system, the next steps will be focused on the addition of more viewers: a general purpose Image Dataset Viewer, and a Viewer specific for the runic text (and its transcription) inscribed in the cross.

The Image Dataset module will be focused on presenting to the user a collection of images/photos, with the ability to browse the dataset and zoom into images. To avoid linking a spot only to a specific point on a single image (which is quite simplistic), we will follow the concept of the PileBars [Brivio et al. 2012], where a thumbnail bar is dynamically rearranged according to some distance metric in the image set (for example, the closeness of the framed subject). In this way, focusing to a spot will reorder the dataset by bringing in foreground all the images framing (or near to) the selected spot.

For the Runic viewer, we will concentrate on the visualization of the various different transcriptions and annotated texts of the runic inscription on the Cross, which is the most important subject of study on the Cross. The idea is to be able to explore the transcription at different levels (sentence, word, rune) while always preserving the connection between the transcription and the signs on the 3D model.

On a longer time scale, as already mentioned in Section 3.5, a proper web-based authoring system will be made available, in order to help the easier creation of the annotated datasets, and the composition of the visualization page.

6 Conclusions

In this paper, we presented a modular, extendible framework for the presentation and navigation of interconnected 3D models and textual information. The user can explore the datasets in a completely free way, or take advantage of a set of reference point (spot) that link to all the modules of the presenter. The spots can be further grouped to easily create lectures following different conceptual paths. A multiresolution implementation enables to visualize and navigate extremely dense 3D models, so that even very complex artworks can be shown and analyzed in realtime.

The system proved to be easy to use and very flexible, since an appealing visualization of a generic object is automatically generated

only by changing the XML content files. The creation of the informative content does not need any previous skill in 3D visualization, since the Webpage is automatically arranged by the module manager.

The system structure was designed to be fully extensible, in order to be able to include every type of information (images, videos, sound...) that could help in enhancing the richness and the interconnections of the dataset describing an object of interest.

Acknowledgements

This work has been partially founded by the European Projects Ariadne (FP7-INFRA-2012-1-313193) and 3D-ICONS (FP7-ICT-PSP-297194). The authors wish to thank the Visionary Cross Project (<http://www.visionarycross.org/>) and, in particular, Prof. Daniel Paul O'Donnell and Prof. Roberto Rosselli Del Turco, who helped with the design of this system.

References

- Microsoft ActiveX Controls. [http://msdn.microsoft.com/en-us/library/aa751968\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa751968(VS.85).aspx).
- BEHR, J., ESCHLER, P., JUNG, Y., AND ZÖLLNER, M. 2009. X3dom: a dom-based html5/x3d integration model. In *Proceedings of the 14th International Conference on 3D Web Technology*, ACM, New York, NY, USA, Web3D '09, 127–135.
- BOWMAN, D. A., NORTH, C., CHEN, J., POLYS, N. F., PYLA, P. S., AND YILMAZ, U. 2003. Information-rich virtual environments: theory, tools, and research agenda. In *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM, New York, NY, USA, VRST '03, 81–90.
- BRIVIO, P., TARINI, M., PONCHIO, F., CIGNONI, P., AND SCOPIGNO, R. 2012. Pilebars: Scalable dynamic thumbnail bars. In *VAST*, 49–56.
- BRUNT, P., 2010. GLGE: WebGL for the lazy. <http://www.glge.org/>.
- CALLIERI, M., ANDREI, R. M., DI BENEDETTO, M., ZOPPÈ, M., AND SCOPIGNO, R. 2010. Visualization methods for molecular studies on the web platform. In *Proceedings of the 15th International Conference on Web 3D Technology*, ACM, New York, NY, USA, Web3D '10, 117–126.
- CALLIERI, M., CHICA, A., DELLEPIANE, M., BESORA, I., CORSINI, M., MOYÉS, J., RANZUGLIA, G., SCOPIGNO, R., AND BRUNET, P. 2011. Multiscale acquisition and presentation of very large artifacts: The case of portalada. *J. Comput. Cult. Herit.* 3, 4 (Apr.), 14:1–14:20.
- CIGNONI, P., GANOVELLI, F., GOBBETTI, E., MARTON, F., PONCHIO, F., AND SCOPIGNO, R. 2005. Batched multi triangulation. In *Proceedings IEEE Visualization*, IEEE Computer Society Press, Conference held in Minneapolis, MI, USA, 207–214.
- CONGOTE, J., SEGURA, A., KABONGO, L., MORENO, A., POSADA, J., AND RUIZ, O. 2011. Interactive visualization of volumetric data with webgl in real-time. In *Proceedings of the 16th International Conference on 3D Web Technology*, ACM, New York, NY, USA, Web3D '11, 137–146.
- DELILLO, B., 2009. WebGLU: A utility library for working with WebGL. <http://webglu.sourceforge.org/>.
- DI BENEDETTO, M., PONCHIO, F., GANOVELLI, F., AND SCOPIGNO, R. 2010. Spidergl: a javascript 3d graphics library

- for next-generation www. In *Proceedings of the 15th International Conference on Web 3D Technology*, ACM, New York, NY, USA, Web3D '10, 165–174.
- DON BRUTZMANN, L. D. 2007. *X3D: Extensible 3D Graphics for Web Authors*. Morgan Kaufmann.
- FARADAY, P., AND SUTCLIFFE, A. 1997. Designing effective multimedia presentations. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, ACM, New York, NY, USA, CHI '97, 272–278.
- GÖTZELMANN, T., VÁZQUEZ, P.-P., HARTMANN, K., NÜRNBERGER, A., AND STROTHOTTE, T. 2007. Correlating text and images: Concept and evaluation. In *Proceedings of the 8th international symposium on Smart Graphics*, Springer-Verlag, Berlin, Heidelberg, SG '07, 97–109.
- HALASZ, F., AND SCHWARTZ, M. 1994. The dexter hypertext reference model. *Commun. ACM* 37, 2 (Feb.), 30–39.
- JANKOWSKI, J., AND DECKER, S. 2012. A dual-mode user interface for accessing 3d content on the world wide web. In *Proceedings of the 21st international conference on World Wide Web*, ACM, New York, NY, USA, WWW '12, 1047–1056.
- JANKOWSKI, J., SAMP, K., IRZYNSKA, I., JOZWOWICZ, M., AND DECKER, S. 2010. 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*, ACM, New York, NY, USA, CHI '10, 1321–1330.
- KAY, L., 2009. SceneJS. <http://www.scenejs.com>.
- KHRONOS GROUP, 2009. Khronos: Open Standards for Media Authoring and Acceleration.
- KHRONOS GROUP, 2009. OpenGL ES - The Standard for Embedded Accelerated 3D Graphics.
- KHRONOS GROUP, 2009. WebGL - OpenGL ES 2.0 for the Web.
- POLYS, N. F., BOWMAN, D. A., AND NORTH, C. 2011. The role of depth and gestalt cues in information-rich virtual environments. *Int. J. Hum.-Comput. Stud.* 69, 1-2 (Jan.), 30–51.
- PRIYOPRADONO, B., AND PERDANA, F. A. 2013. Article: Web3d publishing interior design and residential collection based on webgl technology. *International Journal of Computer Applications* 64, 1 (February), 10–14. Published by Foundation of Computer Science, New York, USA.
- RAGGETT, D. 1995. Extending WWW to support platform independent virtual reality. *Technical Report*.
- SONNET, H., CARPENDALE, S., AND STROTHOTTE, T. 2005. Integration of 3d data and text: the effects of text positioning, connectivity, and visual hints on comprehension. In *Proceedings of the 2005 IFIP TC13 international conference on Human-Computer Interaction*, Springer-Verlag, Berlin, Heidelberg, INTERACT'05, 615–628.
- YANKELOVICH, N., HAAN, B. J., MEYROWITZ, N. K., AND DRUCKER, S. M. 1988. Intermedia: The concept and the construction of a seamless information environment. *Computer* 21, 1 (Jan.), 81–83, 90–96.