X3D/X3DOM, Blender Game Engine and OSG4WEB: open source visualisation for cultural heritage environments

Antonio Baglivo, Francesca Delli Ponti, Daniele De Luca,
Antonella Guidazzoli, Maria Chiara Liguori
CINECA
Casalecchio di Reno (Bologna), Italy
visitlab@cineca.it

Bruno Fanini
CNR-ITABC
Rome, Italy

***Abstract*— Cultural Heritage applications have often exploited 3D reconstructions in order to better convey contents, but virtual environments need also new solutions to be accessed in a flexible as possible way. X3DOM, Blender Game Engine and OSG4WEB are three different tools able to perform on-line, mobile and desktop visualisations, each one with pros and cons. Do they meet the requirements raised by Cultural Heritage curators? This paper will be offering an overview of these navigation tools by means of some case studies in order to better highlight problems and opportunities.**

Keywords—Open Source, Cultural Heritage, Real Time Virtual Environments, X3D, Blender, OpenSceneGraph.

I. INTRODUCTION

The Cultural Heritage and ICT fields have nowadays gathered long years of fruitful collaborations and recently the first one is taking advantage from the spreading of open source solutions. Cultural Heritage applications have often exploited 3D reconstructions in order to better convey contents. Virtual environments can now rely upon some very competitive and effective open source tools, such as the Gimp framework for image processing, Blender for 3D modeling, MeshLab for 3D scanned data editing. Moreover virtual environments need solutions to be accessed in a flexible as possible way, especially now that multiplatform output has become a mostly desirable option for digital applications. X3DOM, Blender Game Engine and OSG4WEB are three different tools capable of performing on-line, mobile and desktop visualization, each one with pros and cons. Do they meet the requirements raised by Cultural Heritage curators? This paper will be offering an overview of these frameworks focusing on some case studies in order to better highlight problems and opportunities offered.

II. OPEN SOURCE AS A VIABLE SOLUTION

Our experience in Cultural Heritage has been developed during several years and many different projects. Along this path we have been focusing more and more on Open Source software, a choice that has greatly influenced our approach to 3D modelling. Until some years ago, the preference for certain modelling software was guided mainly by the kind of output file formats delivered by the different software packages; file formats that, for the specific application we were developing, should have been consistent with real-time navigation within virtual environments. Currently, decisions can follow different paths, since there is a good range of open source tools both for modelling and navigating. The first step was a shift from consolidated 3D modelling softwares, such as Creator [1] and 3dsMax [2], towards rapidly growing open source solutions, such as Blender [3]. The second step focused on the navigation software. For some years a software internally developed at Cineca, Visman [4], was used for real-time desk-top applications and immersive environments for the virtual theatre, but the need for more flexible tools led to experimenting with open source tools, such as X3DOM [5], Blender Game Engine [6] and OSG4WEB [7], jointly developed by Cineca and CNR-ITABC [8, 9].

The integration of open source visualisation tools inside the production pipeline becomes a relevant step forward if Cultural Heritage operators make the decision of following an open source structure/pipeline of production. Maybe encouraged by projects that already make use of open software, or because considering it advisable, for whatever reason, going towards a complete open approach that - following the open data philosophy - foresees the distribution of models besides that of the product, developers of applications dedicated to cultural heritage need also open source visualisation tools, going beyond Unity3D [10], at present one of the preferred ones in CH [some examples in 11, 12].

III. 3D VISUALISATION TOOLS: CULTURAL HERITAGE REQUIREMENTS

In order to have an easy access to Cultural Heritage contents, there are some commonly required features. Besides these, each application might need further custom features. The first requirement is free navigation of the virtual environment and the possibility of automatic tours, to better highlight the most interesting elements. Useful features for 3D models are all those that address better upload and fluent navigation or allow to manage the complexity better: instruments such as LOD (level of detail) or paging are fundamental. These instruments smartly recognize the different parts of the 3D

978-1-4799-3170-5/13/$31.00 ©2013 IEEE

environment and allow to upload and download only the needed parts of the model.

Virtual environments may have also interactive features in order to add information to the mere 3D model so that models are connected to videos, web pages or database with, for example, historical information. The navigation software should recognize the interaction of users, usually a mouse click, and open other windows with further documentation. It is also important a switch option among models, for different purposes depending on the project. In our experience we used switch and story nodes to highlight time phases of building construction or alternative hypotheses.

As said before, every project could have peculiar needs, hence, customizable navigation software is very important.

A. Web3D: from VRML to WebGL and X3DOM

VRML (Virtual Reality Modelling Language) has traced the way for the web based 3D visualization. In 1995 it was unique because it supported 3D geometry, animation, and scripting. In 1997, VRML was ISO certified and continued to attract a large following of artists and developers [13]. The related extension is .wrl and it is a simple textual file through which every part is specified into a scene graph. VRML is also a standard file format (ISO/IEC 14772-1:1997) and it has been particularly designed for the Web. Visualisation is enabled by a specific plug-in or a VRML browser [14]. Some interesting examples of real-time tours in VRML are those developed by Bitmanagement using a plug-in of theirs, BS Contact [15].

On the web, the direct fruition of 3D environments through a browser has been a major challenge: the call of OpenGL functions through browsers has been achieved thanks to WebGL, a JavaScript framework. WebGL is JavaScript library enabling CG programming (real-time rendering): the 3D scene, embedded in web pages, is rendered by the GPU and functions as an OpenGL wrapper. WebGL can render inside browsers both 2D and 3D contents; it is completely integrated inside web standards and exploits GPU acceleration for real-time. Its main disadvantage is the programming effort and the need of implementing a lot of codex for every scene.

In order to overcome this problem, Fraunhofer Institute implemented X3DOM [5], a JavaScript framework combining X3D, WebGL, HTML (5), CSS and JavaScript [16]. X3D is a document describing a scene graph in a declarative way (standard ISO/IEC 19775/19776/19777). The document follows a specific XML grammar [17]. The supported scene graph can handle multiple scenes (or stages), multi-texturing and rendering and supports shaders with normalmap and lightmap. X3DOM embeds the content of a scene graph, described in X3D, inside a web page (without plug-ins). It analyses the content of the X3D file and uses WebGL for the rendering. The browser opens a HTML 5 web page and JavaScript manages and manipulates its elements and those coming from other XML documents, such as X3D documents. Using the X3DOM framework we can embed the content of an X3D document and, interacting at run-time with the content, produce both animations and reactions inside the environment according to users' created events.

The absence of a plug-in is not the only advantage of X3DOM; as a JavaScript framework it can enable three kinds of interactions:

- an event in the scene that causes a behaviour in the scene;
- an event in HTML5 that causes a behaviour in scene. The event can be a button clicked or a mouse over specific areas of the HTML5 page. These events can trigger real-time behaviours inside the scene, such as the starting of animations or a camera movement;
- an event in the scene that causes a behaviour in HTML5 page.

The complete freedom left to developers to catch and manage every kind of event has led to innovative projects [18]. The only need is an updated browser with WebGL integrated. The Web home page of X3DOM has a test browser page to verify the compatibility [19].

B. Blender - Game Engine

Since the 2.37 version, Blender modelling software has a complete internal game engine (Blender Game Engine - BGE) enabling the creation of 3D interactive applications. BGE is a high level programming tool for game development but it can be profitably employed for the creation of whatever 3D interactive application, such as architectonic or archaeological interactive tours.

The core of the BGE structure is in the so called Logic Bricks, making available an easy to use visual interface for the creation of interactive applications that can function without any knowledge of programming languages. There are three typologies of Logic Bricks: Sensors, Controllers and Actuators. Otherwise, the Game Engine has its specific [Python API], separated by Blender itself, for the creation of videogames in Python and for scripting customised interactions. This option can be activated through the logic brick "controller" called Python, which enables the link of scripts in Python. BGE enables the possibility of controlling shaders and textures, visualised during modelling creation with the 3D view; materials/shaders are visualised as singletexture, multitexture and GLSL, which uses the OpenGL Shading Language. Therefore it is possible to use real-time lighting, bump and normal maps and real-time reflections. Blender is, first of all, an advanced 3D modelling software but, thanks to its internal game engine, interactions scripts and world rules, such as gravity, can be defined and activated by the logic bricks or the Python scripts. By pressing the key "P", scenes can be launched directly inside the 3D viewer, tested and published without further ado.

In Blender, scenes are, if not specified otherwise, independent and define different locations with different lighting sets. This kind of organisation is devised mainly for film production, mimicking through the usage of scenes and shots, with locations and framings, the conventional production pipeline of movies and short films. This system can be exploited for real-time navigations and explorations: scenes enable the division among locations and different world configurations. Each scene can have its own models, locations, physical properties, logic bricks and *ad-hoc* scripts.

For the production of Cultural Heritage contents, the scene subdivision system enables the use of Actuator logic bricks for overlaying or replacing models in order to convey further information or a different historical period. This option is assimilable to the functions Switch and Story, so relevant for this kind of projects.

The strong support delivered by the Blender community, that releases tools and scripts following the same open philosophy of the software, allows the continuous integration of new functionalities thanks to Python language and external libraries. Some interesting and recent examples are the support given to Leap motion [20], Kinect [21] and 3D mouses, as well as the support and integration of software protocols, such as OSC-MIDI, used for data transmission and interpretation of data also arriving from smartphones and tablet with sensors, such as gyroscopes, accelerometers and compasses, that can be useful for 3D interaction. Thanks to these technologies, real-time navigations can be programmed for a more natural interaction and immersive feeling.

Through the complete integration of OpenGL, Blender supports inside the game engine different modalities of stereoscopic visualisation:

- OpenGL Quad-Buffered: generation of two images with different frame-rates;
- Above-Below: the two images are shown one above the other;
- Interlaced: the two images are rendered by alternate rows;
- v-interlaced: the two images are rendered by alternate columns;
- Side by Side: rendering alongside image;
- Anaglyph: both images are rendered and superimposed with a red-cyan anaglyph.

The above mentioned modalities can be applied to an automatically generated stereo camera and managed with the Eye Separation parameter that, in meters, sets the two virtual views. Further setups are possible with Python scripting on the dedicate functions setEyeSeparation() and setFocalLength(), enabling also the variation of the stereo effect intensity in real-time.

Logic Bricks and/or Python scripts implement the navigation inside the 3D environments through manual navigation and/or scripted pre-defined camera movements, for one or more guided tours. Since 2.5-2.6 version, Blender includes automated algorithm for path finding and the creation of navigation mesh, conceived for game implementation but suitable also for the creation of systems with automatic navigation /artificial intelligence.

1) *Burster:* Burster is an open-source multi-platform plug-in, available at present for Windows and Linux [22]. The gameplay is safely integrated in web sites through sandboxing and optional support to cryptography. As a visualisation plug-in for contents developed with BGE, it owns all the features of the engine, such as the support to BGE Python, Logic Bricks, scenes/levels and GLSL visualisation modalities. In order to visualise games and navigations on the web, there are some more passages besides those needed with a classic BGE file, such as:

- devise the blend file with all the libraries needed or incorporate them before the deploy;
- pack textures and all the external associated file into the main blend file in order to have only one final blend file;
- create a web page calling both the blend file and Burster using embedded tag, as indicated on the Burster page.

Burster's plus is in the possibility of integrating real-time navigation, games and interactivity inside web pages, and there is no need to create any new code since it exploits all BGE features. Disadvantages are in the necessity to pack everything in just one file, losing the option of selective downloads of libraries and, consequently, the possibility of paging 3D scenes, mismanaging the computing power of the machine and the band width. Another disadvantage comes from the impossibility of including some external Python libraries, not installed for the web plug-in, and of calling HTTP.client, unless compressing the file with the cryptographic algorithm available for a fee on Burster web page.

2) *Blender Player for Android:* Even if as an independent feature from the Blender release, a project pertaining to GSoC 2012 has adapted BGE to Android, enabling the access on mobile platforms of real-time contents created for the game engine. This player cannot count on a continuous support but the present version already performs Blender files with features similar to those for desktops, supporting: Python scripting, logic bricks, shaders single and multitexture, alpha channel for textures or texts, multiple scenes. At the moment there are showstoppers such as the lack of: dynamical texts, with or without text mesh, multi touch, debugging tools, access to internet or to Android file system.

C. OpenSceneGraph and OSG4WEB plug-in

Real-time rendering of virtual worlds for Cultural Heritage purposes has quite a demand nowadays. The OpenSceneGraph framework, being one of the best 3D graphics toolkits, is being widely used in the fields of VR, scientific visualization, visual simulation, modelling, games, mobile applications and others [23].

Through past experiences, such as the "Virtual Rome" project [24], a custom 3D web plug-in OSG4WEB [7] based on the open source framework OpenSceneGraph [25] was developed by Cineca and CNR-ITABC and subsequently extended with further features, added in response to application requirements. The diverse and growing community of over 5,000 developers for the efficient OpenSceneGraph framework is centred around the public osg-users mailing list, where members discuss how to use OpenSceneGraph at best, provide mutual support, and coordinate development of new features and bug fixes. Within "Aquae Patavinae" case study [26], the renewed OSG4WEB plug-in [27] has capabilities of advanced lighting, such as self-shadowing in real-time, provides a customizable 3D interface, an enhanced navigation system and the action scripting within the virtual world. The capability of embedding action scripts inside 3D objects allows the construction of action graphs to customise user interactions, for instance activating different interpretations and reconstructions of archaeological sites [28]. The scene can be managed by a web-based Back-End that allows a restricted team to manage and modify 3D models in a library, while a customizable Front-End component (PHP) allows the real-time exploration of the published landscape. The OpenSceneGraph integration showed consistent performances especially with large scenes, where the system excels through the use of a state-of-the-art database pager to manage landscape segmentation and 3D content streaming to final users, even with the enrichment of massive 3D vegetation (Fig. 1, lower row). On the other hand, the plug-in showed also good adaption to indoor scenes (Fig. 1, upper row: reuse of Villa di Livia datasets from Virtual Rome into the new plug-in).

Fig. 1. Visualization examples through OSG4WEB plug-in.

Special attention was also dedicated to the navigation system, which evolved alongside the integration of the plug-in with natural interaction systems [29]. The two natural interactions deployed into a few public exhibits, acted as different layers to reach the virtual world content, with the following objectives:

- Touch-screen: provide a smooth experience for the user during the exploration of valuable hotspots;
- Body Tracking: provide a low-cost solution to track visitor, without any equipment or assumptions regarding the user.

Future developments regarding public installations will concentrate on the results obtained during user tests, surveys and observations, to provide an enhanced visitor involvement while using the plug-in and to deliver an experience during the exploration of large and rich landscapes within virtual heritage applications.

IV. A VERSATILE TOOL FOR A WIDE AUDIENCE: BLENDER 3D

A short introduction to Blender is due, before starting with the overview on how Blender models fit to the different open source viewers.

Blender software, conceived in 1993 and available since 1994, has become one of the most popular 3D Open Source applications and delivers 2D and 3D contents (3D visualisations, static images, quality videos for TV and cinema, 3D interactive contents) offering a wide range of features, such as modelling, texturing, lighting, animation and video post processing. Its open architecture offers cross-platform interoperability, extensibility, small dimensions and an integrated working pipeline.

Originally produced by the company "Not a Number" (NaN), Blender is now a "Free Software", its source code available under GNU GPL licence, developed by the Dutch Blender Foundation. The latest version is 2.67b (May 2013).

A. Blender for modelling and open source: porting of models

An advanced, Open Source 3D modelling tool, Blender's management and simplification of scenes with high polygon counts can help with rendering animation, speed and photorealism. Its manifold opportunities had led to the creation several importers and exporters. Among the different possibilities there is also the export of laser scanning acquisitions from MeshLab, hence, importing point clouds or meshes in Blender without losing information. In Blender, meshes can be better texturized for rendered videos or can be optimized and simplified through decimation and baking of the textures, suitable for real-time applications. Among the other there are OBJ, X3D, FBX [30].

V. CASE STUDIES AND EXPERIMENTATIONS

In the previous paragraphs we have presented an overview of open 3D visualisation tools, besides a brief presentation of the main 3D modelling software. For each one of them, we are now presenting case studies used for testing the tools and exploring their potentialities.

A. From Blender to X3D

Nowadays the importance of the web in spreading 3D models has incredibly grown. X3D, plus X3DOM, has drawn our attention because, from a first comparison, performances proved to be high. Therefore we began extensive testing on the features already available and their potentialities.

Our first case study for the X3D technology was the porting of the reconstruction of the Third Cloister pertaining to the monumental cemetery in Bologna. The virtual environment was implemented several years ago by using 3dsMax and, at present, is visible on a web site using a downloadable plug-in [31, 32]. First of all we converted the original model to Blender and then tested it on the web using the X3D technology.

This model is going to be part of Bologna's Open Data repository [33] and the Blender open format can be read without any license problem.

The Blender model was exported in the X3D format in order to compose a web page so that the models could be freely navigated. Blender already has an exporter that allows the creation of X3D files directly from Blender models: the exporter enables the creation of X3D files of selected Blender objects so it is possible to export single sections in order to arrange the 3D models directly on the web page and, in case, to add specific X3D nodes.

As said before, navigation software for Cultural Heritage environments needs to have at least the following features:

- free navigation and defined automatic tours;
- connection of models to database or further information;
- switch among different models.

All these features are available in X3D. The free navigation is already defined by rules [34] but it is also possible to add pre-defined points of view or automatic tours. Inside the HTML page the tag *viewpoint* defines a point of view in terms of position and other characteristics necessary for positioning a camera. Other tags, besides viewpoint, are already available, but further implementations are possible by using JavaScript. For example, the model of the Third Cloister has some buttons defined with JavaScript to call different cameras or an automatic tour inside the virtual environment (Fig. 2).

The Third Cloister, with its array of relevant artistic tombs, is linked to an on-line database. The relationship between models and the database was implemented by using an existing tag called *anchor*. This tag defines an interaction with some objects during the virtual tour and with a link to be opened in case of its activation.

Every X3D created object is identified with a tag called *group*, specifying the name of the object that is the same of its corresponding Blender object. Note that inside Blender it is possible to create real groups, made of different objects, but these entities could not be exported into X3D. The exporter recognizes Blender instances: inside the X3D file they are identified in the *group* tag, with the parameter *use*, that calls the original instanced object with its name.

Fig. 2. Third Cloister: X3D Viewpoint buttons, Uttini's tomb and related database card

Our second case study using X3D technology is about a Roman villa, called "Villa di Russi" [35]. This latter case is part of PARSJAD project (Archaeological Park of the High Adriatic) [36]. The project aims at highlighting the cultural and archaeological heritage of the territory between North-East of Italy and Slovenia and consists in the development of a prototype capable of reading heterogeneous repositories and harvesting the data. Data are interpreted and shown through the graphic interface of Google Earth (or Google Maps in a Linux or Android environment), with placemarks containing a descriptive balloon and an automatically generated explanatory file. As an option and when available, there could be photo overlay and/or low poly 3D models in Google Earth Interface or High Poly X3D Models with the pertaining metadata showed in a separate link. "Villa di Russi" is an instance of this graphics visualisation. The model is built in Blender, exported in X3D and navigated on a web page thanks to X3DOM framework. The challenge is to associate some areas of the Villa with metadata containing their descriptions: this metadata is read through an XML file located on server.

Fig. 3. Web interface for the Villa di Russi application.

The web interface is divided in four parts: the main part is the 3D scene, interpreted by X3DOM framework, on the lower side there are some buttons for changing the point of view of the camera and for framing and highlighting areas of interest; it is possible to get the same behaviour by means of the mini-map located in the panel on the right side; clicking either the area of interest in the scene, the button or the mini-map, in the right panel on top appears a metadata description: these metadata are parsed from XML files located in server side (Fig. 3).

B. From Blender to Blender Game Engine and Burster

Blender Game Engine (BGE) allows the creation of different types of navigation, the camera system allows the use of a guided automatic navigation, as in cinematic scenes in games, but also to implement a "follow" constraint in order to force the camera to follow an avatar, thus creating a third person navigation.

An interactive and configurable simulation of Cineca Virtual Theatre was used for testing the BGE. The application redesigns and reconfigures the theatre, allowing navigation in the structure and switching among different configurations for an interactive architectural visualization. AutoCAD DXF and DWG files were the base for the modelling process, mixed with new possible options for the set-up of the theatre.

For the implementation of the real-time application, procedural textures and materials built for the rendering, not supported by the game engine, had to be baked. The high-poly models had to be re-modelled in low-poly or underwent modifiers such as *Decimate* and *Remesh*, automatically reducing complexity while protecting dimensions and geometries. High-poly objects were baked on the low-poly ones, including texture baking, normal maps and scene lighting (Fig. 4). For 3D navigation a freely available script was adapted [37]. It implements movement through keyboard and mouselook (not automatic first person navigation). From the

Logic Bricks we used a *Near* sensor to trigger the switches between geometries whenever the viewer approaches them. Another Python script activates a video, set on a screen wall.

At present the demo is visible off-line on desktops and as a stereoscopic application for the Virtual Theatre. For the 3D stereo view, a specific BlenderPlayer launcher with QuadBuffered mode was implemented.

Fig. 4. Real time and interactive architectural visualisation made with the Blender Game Engine.

Another demo was implemented with Burster plug-in, showing the photovoltaic car park at Cineca (Fig. 5). Semi-transparent billboard backdrops integrated the car park inside the wider context. A digital signage system for data monitoring was integrated inside the application. An XML file, delivered from the PV management system with real-time digits, allows us to show data inside the 3D environment. Two Python scripts (called HttpFetch and HttpDisplay) use an XML DOM to interpret the XML document and locally store the values and create a dynamic "text" object, instantiated and populated with freshly extracted values from the XML files.

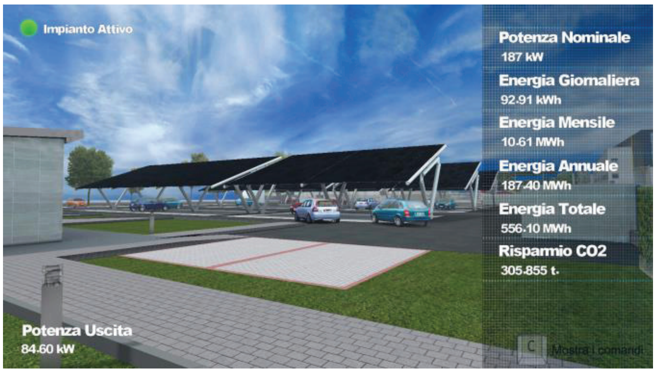

Fig. 5. Interactive visualization with real-time data in the Blender Game Engine

The on-line navigation with Burster required some special precautions: texture optimization; application of all Blender modifiers; dynamic lighting was disabled inside the scene that uses OpenGL Shading Language (GLSL), therefore, environmental lighting is baked in the textures. In order to function independently, the blend file has to encompass textures, data and models; in order to communicate with on-line resources, such as the data coming from the XML file, the BursterEncode tool or an upload on the Internet site of the project enabled an encryption phase, during which .blend source file was processed and encrypted.

C. From Blender to OSG4WEB

The workflow from Blender to OSG4WEB has been refined mainly through the experience of some projects and case studies like "Aquae Patavinae" [26] and "osgCrowd", for simulating large crowds in large cities. A typical pipeline (Fig. 6) steps through data acquisition, 3D modelling and asset production and, finally, data organization and optimization for online fruition through the plug-in or a simple OpenSceneGraph-based viewer.

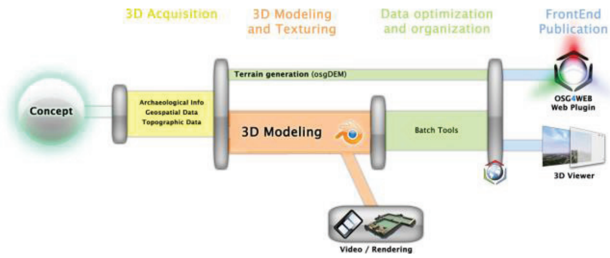

Fig. 6. Production pipeline from concept through Blender to OSG4WEB.

Regarding terrain generation, the open-source "Virtual Planet Builder" project [38] provides tools such as osgDEM for creating well-balanced large scale paged databases from geospatial imagery and digital elevation maps. On the other hand, a set of offline and server-side tools have been developed [27] in conjunction with Blender modelers with the following purposes:

- to provide automated process to build balanced hierarchies (scene-graphs) for optimal web streaming (e.g. large cities);
- to provide automated spatial segmentation using native OpenSceneGraph pager;
- to provide tools for texture compression, geometry and material optimization;
- to provide remote server-side tools to generate native OpenSceneGraph instancing, given a template model and Blender-generated instance-list.

Since digital representations of virtual environments have constantly increased their complexity, these tools are able to produce complex and detailed 3D contents able to be streamed via web and also to maintain an interactive frame rate. This workflow, involving Blender and OpenSceneGraph, has been successfully applied also to visualise and efficiently manage large 3D crowds within a case study of urban mobility in the centre of Bologna city [39].

Fig. 7. Imported Bologna center from Blender and real-time visualization of simulated crowds.

Existing open-data assets produced within APA and NuME projects [40, 41] have been re-used to generate a portion of the city centre (Fig. 7), automatically imported and spatially segmented, alongside with direct exportation from Blender to OpenSceneGraph format of a few 3D animated avatars, used to represent virtual humans.

VI. Conclusions

The following comparative table summarises some of the features that can be relevant to Cultural Heritage 3D environments.

TABLE I. VISUALISATION TOOLS AND RELATED FEATURES - SYNOPTIC TABLE

	X3D / X3DOM	Blender Game Engine	OSG4WEB
Geometry instancing	✓	✓	✓
External references	✓	✓	✓
LOD	✓	✓	✓
Multitexture	✓	✓	✓
Switch	x3d: ✓ x3dom: ✗	✓	✓
Billboard	✓	✓	✓
Shaders	✓	✓	✓ (GLSL)
Viewpoint automatic navigation	✓	✗	✓
Stereo	✗	✓	✓
Fog	✓	✓	✓
External peripherals support	x3d: ✗ x3dom: ✓ with javascript	✓	✓
Web compatibility	✓	plug-in	plug-in
3D Text	✓	✓	✓
Grouping Nodes	✓	✓	✓
Sound	✓	✓	✗
Paging (large scenes)	✗	✗	✓
Multiple Scenes	✓	✓	✓
Hierarchical Culling	✓	✗	✓
Occlusion Culling	✓ [42]	✓	✓

This state-of-the-art overview demonstrates that X3D is a flexible solution, both for on-line desktop and mobile applications Users are not supposed to download any plug-in, but it is important to install updated browsers with WebGL integrated, a mandatory requirement for 3D environments. X3D is a good platform, but currently for virtual reconstruction of terrains OSG4WEB seems the best choice. Nevertheless, OSG4WEB is a demanding tool, since advanced OpenGL skills are required for programming the scene and a plug-in installation is needed for visualisation. Blender Game Engine and Burster can be collocated in-between the other two visualisation tools. On the one hand BGE is positively incorporated into its own 3D content editor, enabling an easy set-up of interactions and a faithful reproduction of the overall performance and behaviour of real-time rendering of the scene. On the other hand, the transition towards the Web is a little bit hampered by the need of a plug-in and a series of tiny arrangements in the creation of the final blend file. Both X3D and BGE/Burster can be used also by users without specific programming skills.

These three open source solutions witness the good quality level achieved, improvements are continuously on going so that they can be considered as a valuable alternative to commercial visualisation tools.

Acknowledgment

The authors wish to thank Luigi Calori, Silvano Imboden, Micaela Spigarolo, Cineca; Sofia Pescarin, Guido Lucci Baldassari, Daniele Ferdani and Emanuel Demetrescu, CNR ITABC; Antonella Coralini, Bologna University; Fiamma Lenzi, IBC; Marina Marcosignori; Mauro Felicori, Comune di Bologna.

v-must.net

The research leading to these results is partly funded under the European Community's Seventh Framework Programme [FP7 - 2007/2013] under the Grant Agreement 270404

References

[1] Creator (http://www.presagis.com/products_services/products/modeling-simulation/content_creation/creator/)

[2] 3dsMax (http://www.autodesk.com/products/autodesk-3ds-max)

[3] Blender (http://www.blender.org/)

[4] C. Borgatti, L. Calori, T. Diamanti, M. Felicori, A. Guidazzoli, M.C. Liguori, M.A. Mauri, S. Pescarin, L. Valentini, "Databases and Virtual Environments: a Good Match for Communicating Complex Cultural Sites", in Proceeding of: ACM SIGGRAPH, Los Angeles, 2004.

[5] X3DOM (www.x3dom.org)

[6] Blender Game Engine (http://www.blender.org/education-help/tutorials/game-engine/)

[7] OSG4WEB (http://147.162.44.42/public/osg4web/osg4web.htm)

[8] S. Pescarin, M. Forte, L. Calori, C. Camporesi, A. Guidazzoli, S. Imboden, "Open Heritage: an Open Source approach to 3d real-time and web-based landscape reconstruction", in VSMM2005,

Proceedings of the 11th International Conference on Virtual Systems and Multimedia: Virtual Reality at Work, 2005.

[9] G. Lucci Baldassari, “Il caso di studio della Villa di Livia per un’analisi comparativa di differenti tecnologie per la visualizzazione di contenuti 3D nel world wide web”, in Proceedings MIMOS 2012, Roma.

[10] Unity3D (http://unity3d.com/)

[11] A. Merlo, L. Dalco, F. Fantini, "Game engine for Cultural Heritage: New opportunities in the relation between simplified models and database", in Virtual Systems and Multimedia (VSMM), 2012 18th International Conference, pp.623,628, 2-5 Sept. 2012, doi: 10.1109/VSMM.2012.6365993

[12] E. Pietroni, C. Rufa, “Natural interaction in Virtual Environments for Cultural Heritage: Giotto in 3D and Etruscanning study cases”, in Virtual Archaeology Review, Volumen 3, Número 7, diciembre 2012.

[13] VRML (http://www.web3d.org/realtime-3d/x3d-vrml/x3d-vrml-most-widely-used-3d-formats)

[14] VRML browser (http://www.web3d.org/x3d/vrml/tools/viewers_and_browsers/)

[15] VRML demos by Bitmanagement (http://www.bitmanagement.de/en/showcase/interactive-demos)

[16] Y. Jung, J. Behr, H. Graf, “X3DOM as Carrier of the Virtual Heritage”, in International Workshop 3D-ARCH, 4, Trento, 2011.

[17] XML grammar (http://www.web3d.org/specifications/x3d-3.0.xsd)

[18] M. Cabral, M.K. Zuffo, S. Ghirotti, O. Belloc, L. Nomura, M. Nagamura, F. Andrade, R. Faria, L. Ferraz, “An experience using X3D for Virtual Cultural Heritage”, in Proceeding of the Twelfth International Conference on 3D Web Technology, Web3D 2007, pp. 161–164, 2007.

[19] X3DOM compatibility test browser page (http://www.x3dom.org/check)

[20] Leap motion (https://www.leapmotion.com/)

[21] Kinect (www.xbox.com/Kinect)

[22] Burster plug-in (geta3d.com)

[23] R. Wang, X. Qian X., "OpenSceneGraph 3.0", 2010.

[24] M. Forte, S. Pescarin, C. Camporesi, B. Fanini, L. Calori, V. Vassallo, A. Palombini, “Virtual Rome”, in CAA Proceedings, 2009.

[25] OpenSceneGraph (http://www.openscenegraph.com/)

[26] “Aquae Patavinae” project (http://www.aquaepatavinae.it/)

[27] B. Fanini, L. Calori, D. Ferdani, S. Pescarin, "Interactive 3D Landscapes Online", in International Archive of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII-5/W16, 2011.

[28] S. Pescarin, D. Ferdani, B. Fanini, I. Cerato, G.L. Baldassari, "Handling transparency in 3D reconstructed on line environments: Aquae Patavinae VR case study", in CAA Proceedings, 2012.

[29] B. Fanini, “Development and Evolution of Natural Interfaces in Virtual Heritage Applications", in Proceedings of MIMOS Conference, Rome, 2012.

[30] Importers and Exporters for 3D models (http://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Import-Export)

[31] Certosa Virtual Museum (http://www.certosadibologna.it/)

[32] L. Calori, T. Diamanti, A. Guidazzoli, M.C. Liguori, A. Mauri, L. Valentini, “Certosa virtual museum: a dynamic multilevel desktop VR application”, in Poster at Eurographics 2003, 1-6 September 2003, Granada, Spain, 2003.

[33] Bologna’s Open Data repository (http://dati.comune.bologna.it/)

[34] X3D rules (http://x3dom.org/docs/dev/navigation.html#interactive-camera-movement)

[35] PARSJAD project, the “Villa di Russi” (http://3d-test.cineca.it/files/PARSJAD/public/applications/x3d-test/VillaDiRussi2/def3/)

[36] A. Coralini, A. Guidazzoli, F. Lenzi, M. Spigarolo, A. Baglivo, M.C. Liguori, “A Google Cloud Approach to Implement a Graphical Data Access Interface Binding Heterogeneous Cultural Repositories”, in M. Ioannides, D. Fritsch, R. Caffo, R. Davies, F. Remondino, J. Leisner (Eds.), EuroMed2012 - Cultural Heritage, Springer, 2012.

[37] Blender Game Engine scripts for a walk-to interface (http://blenderartists.org/forum/showthread.php?209688-Box-Projected-Cube-Environment-Mapping-demo-(Shiny!))

[38] “Virtual Planet Builder” project (http://www.openscenegraph.org/index.php/documentation/tools/virtual-planet-builder)

[39] Crowd simulation case study (video: http://www.youtube.com/watch?v=J6DzP0MGv7w)

[40] G. Boetto Cohen, L. Calori, F. Delli Ponti, T. Diamanti, A. Guidazzoli, S. Imboden, M.C. Liguori, A. Mauri, A. Negri, S. Pescarin, “Apa the Etruscan and 2700 years of Bolognese History”, in ACM SIGGRAPH ASIA 2011, Posters and Sketches Proceedings, Hong Kong, 2011.

[41] A. Guidazzoli, M.E. Bonfigli, “The creation of the NUME project”, in ACM SIGGRAPH 1999, Los Angeles, 1999.

[42] X3D rules for occlusion culling https://x3dom.readthedocs.org/en/latest/tutorial/culling.html