INTERACTIVE 3D LANDSCAPES ON LINE

B. Fanini[b], L. Calori[a], D. Ferdani[b], S. Pescarin [b]

[a] CINECA, via Magnanelli 5/2, Casalecchio di Reno (BO); l.calori@cineca.it
[b] CNR ITABC, via Salaria km 29,300, 00015 Monterotondo St. (Rome); bruno.fanini@gmail.com, daniele.ferdani@gmail.com, sofia.pescarin@itabc.cnr.it

Commission VI, WG VI/4

KEY WORDS: Web Applications, Web Graphics, OpenSceneGraph, Virtual Museums, Real-Time Graphics, Landscape reconstruction

ABSTRACT:
The paper describes challenges identified while developing browser embedded 3D landscape rendering applications, our current approach and work-flow and how recent development in browser technologies could affect. All the data, even if processed by optimization and decimation tools, result in very huge databases that require paging, streaming and Level-of-Detail techniques to be implemented to allow remote web based real time fruition. Our approach has been to select an open source scene-graph based visual simulation library with sufficient performance and flexibility and adapt it to the web by providing a browser plug-in. Within the current Montegrotto VR Project, content produced with new pipelines has been integrated. The whole Montegrotto Town has been generated procedurally by CityEngine. We used this procedural approach, based on algorithms and procedures because it is particularly functional to create extensive and credible urban reconstructions. To create the archaeological sites we used optimized mesh acquired with laser scanning and photogrammetry techniques whereas to realize the 3D reconstructions of the main historical buildings we adopted computer-graphic software like blender and 3ds Max. At the final stage, semi-automatic tools have been developed and used up to prepare and clusterise 3D models and scene graph routes for web publishing. Vegetation generators have also been used with the goal of populating the virtual scene to enhance the user perceived realism during the navigation experience. After the description of 3D modelling and optimization techniques, the paper will focus and discuss its results and expectations.

1. INTRODUCTION AND DEFINITION OF THE PROBLEM

1.1 Cultural Heritage applications at landscape scale

The paper will describe challenges found while developing browser embedded 3D landscape rendering applications, our current approach and work-flow and how recent development in browser technologies could affect.

In landscape archaeology it is crucial to have an overview of the landscape at large and medium scale at the same time. This is because some phenomena, including relationship among sites, can be just seen analysing the whole context, while others are much more related to a single site. Moreover landscape interpretation and reconstruction requires the cooperation of several disciplines and the adoption of different approaches required by geomorphology, paleo-environmental studies, history, archaeology, topography, ecology etc. That's why recently several studies in this domain have various professionals and researchers working in the same team (Mozzi et al. in print). This is why it would be very important to develop real-time 3d applications where advanced users can explore dynamically a territory, interacting with it, with 3d sites acquired on the field, with database and with other raw data or information. Moreover, since interpretation can greatly benefit from the contribution of several experts in different domains, on line cooperative environments can be a real challenge for the future of landscape archaeology. Landscape Virtual Museum, applications dedicated to a wider public and focused on landscape exploration, can only based on serious interpretation work, while interactivity, narrativity, game mechanisms, can turn a simple 3d exploration into a real experience.

Developing application in CH domain dealing with reconstruction at landscape scale, poses peculiar requirement to rendering engines: there is a huge variation in model scale and detail, navigation interface has to adapt to the scale (fly, walk) content could come from very different pipelines (data acquisition, Computer Graphic-based modelling, procedural) according to time and provenance: for actual status we could get large scale GIS data (DTM and aerial-satellite photos), medium range data (Lidar) as well as detailed models from laser scan cloud points or photo modelling; for past time reconstruction hypotheses, some models are reconstructed by architectural modelling, while at a larger scale (urban level) use of a procedural approach of tools like CityEngine is mandatory to approach larger area reconstructions.

All these data, even if processed by optimization and decimation tools, result in very huge data-set that require paging, streaming and dynamic LOD to be implemented to allow remote web based real time fruition (we assume that remote visualization approaches is not feasible).

One of the most successful example of this kind of applications is Google Earth plug-in that, unfortunately, does not seem to expose a sufficient degree of flexibility to be used as a general purpose development platform.

1.2 State of Art and Previous work

The application area of interactive 3D applications and web based applications have been in touch since the VRML era and his successor X3D were declarative mark-up languages aimed at providing a standardized way to specify 3D content for 3D visualization over the web and within the browser. There were several vendor platform aiming at providing “browsers” for VRML and after X3D content but unfortunately this standard was not able to pass through a process of browser war, standardization and wide adoption that HTML have passed. So, up to now, we do not have a really well established 3D counterpart of HTML. Analysing the many reasons behind standardization failure is out of scope here, we just say that, at present, there are different approaches to the problem complex Web3D application poses and still no single solution has reached a sufficient wide adoption to become a (de facto) standard.

The many solutions could be evaluated by several qualitative as well as quantitative parameters:

- rendering performance (visual quality and frame rate, given a certain HW platform),
- network performance (waiting time to reach the optimal visual quality given a certain bandwidth) flexibility (how many Web 3d applications types can be supported),
- deployability (how easy is the system deployable in the present web, which hardware it can run on).

We can try so cluster systems into the following groups:

- Applications tailored to a specific problems that have been “wrapped” as wrapper plug-ins, one of the most successful is Google Earth, being strictly tied with its huge quad-tree data-base, it is at the top possible rendering and network performance in the field of terrain rendering; Google has managed to wrap GE into a multi-platform, multi-browser plug-in with a rich javascript API. Recently the engine has been enhanced to allow visualizing building data and vegetation. Being closed source and strictly tied to Google Earth database, exposes a limited degree of configurability. There are other commercial products of this type, that are tuned for terrain rendering.
- Systems that wrap a general purpose (game) rendering engine into a browser plug-in, one of the most successful example is Unity which provides a flexible and performance game engine in a software dev platform able to build desktop, web and mobile app. The rendering performance of this kind of web app is usually the highest; unfortunately, they tend to be biased towards gaming applications where database can be tuned to fit game logic; They are less suited for landscape and terrain applications as they often do not support paging out of the box.
- Another important class of web3d apps are the VRML-X3D browsers [X3D], under the form of application or browser plug-ins. They have the advantage of layering on a well defined standard spec for data layer (VRML-X3D) and scripting API. The rendering performance of these systems lags a little behind what available in both game engines as well as custom apps.
- Since 2006, there is also an effort to directly map OpenGL (ES 2.0) into Javascript within browsers [webGL]. WebGL is currently available on development version of all the main browsers excluding IE, and recently included in Chrome release. Being included by most of the browsers, this standard drew a great attention and many demo applications as well as frameworks have been rapidly developed. Being a wrapping of very low level OpenGL, in principle it can be used to address all the possible web app fields.

Higher level abstraction frameworks such as those implementing SceneGraph paradigm are actively developed such as SceneJS, that store the scene graph in JSON format or the X3DOM project that integrates X3D and WebGL [X3DOM].

Some of the WebGL based applications have rendering and network performance that pair with custom developed app, unfortunately we are not aware of any landscape terrain app yet.

Back at the beginning of our OSG4WEB development, there were very few web application development environment delivering the mix of high performance rendering of both terrain as well as interiors required by the main target project (Virtual Rome) (Calori, Camporesi, Pescarin, 2009). Thus we decided to approach the problem by selecting an open source scene-graph based visual simulation library that expose sufficient performance and flexibility and adapt it to the web use by embedding into a browser plug-in.

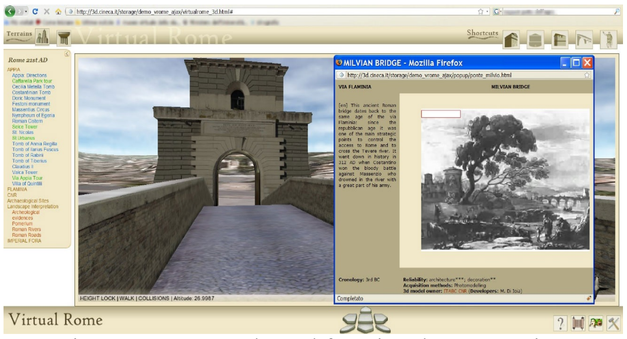
OpenSceneGraph is one of the most used Scene Graph libraries available with open source licensing: it has his main focus in Visual Simulation applications.

This library address both the back and as well as the front end components needs: OpenSceneGraph with its related projects VirtualPlanetBuilder, provide support for terrain rendering pipeline OOB; on the back end side, the OSGdem application allow for quad-tree like hierarchy generation of terrain LOD pages, on the front end side, the scene graph core support paged LOD nodes dynamic loading through http protocol. Furthermore the wide range of file loaders and importers, allowed quite easy integration of modelling pipelines based on Max and Blender.

This way, the format of the data that passes over the wire, can be optimized and tailored to the application needs while preserving data integrity and portability (due to the open source nature of the platform).

By layering on a strong open source SDK, our development effort thus focused on:

1. Browser embedding
2. Application specific front end features
3. Back end tools for model generation, conversion and optimisation



VIA FLAMINIA

MILVIAN BRIDGE

[en] This ancient Roman bridge dates back to the same age of the via Flaminia: since the repubblican age it was one of the main strategic points to control the access to Rome and to cross the Tevere river. It went down in history in 312 AD when Costantino won the bloody battle against Massenzio who drowned in the river with a great part of his army.

Cronology: 3rd BC

Reliability: architecture***; decoration**
Acquisition methods: Photomodelling
3d model owner: ITABC CNR (**Developers:** M. Di Ioia)

Figure 1. OSG4Web used for Virtual Rome Project

The first large scale prototype of this approach has been the Virtual Rome project: we wrapped up a customized OSG based application (viewer) into a Firefox plug-in

("OSG4WEB"); it supported terrain exploration, scene switching, viewpoint and path definition, labelling and picking.

The plug-in has been deployed into an Ajax based web interface. On the back end side, a restricted team have been provided a web interface allowing simple scene updates using appropriate credentials (user name and password) for access. Through this WebLAB tool, a 3D interactive environment is created (Front-End) for large archaeological landscapes exploration on the web. This provides users a rich scene exploration, connecting the GIS functions to the Virtual Reality capabilities, interacting with it, acquiring information etc.

1.3 OSG4Web Plugin

Through the Virtual Rome experience, this platform has been extended and enhanced by adding features to the Front-End plug-in such as real time self shadowing, 3D interface, enhanced navigation and customisable event triggers.

Alongside improved walk mode and navigation behaviours, recently introduced OpenSceneGraph features made possible the adoption of efficient algorithms for real-time shadows, in this particular case, providing a good support for the delicate self-shadowing issue and the tuning on large environments. The primary light source (the sun) in fact, can now be positioned in an interactive way during the virtual world exploration, allowing smooth transitions and possibly simulating time-slices. Shaders and Light-Space perspective shadow-maps techniques also adapt to animated (eg. crowds) and paged content, with aesthetically good results and minor impact on performances for a more immersive user experience.

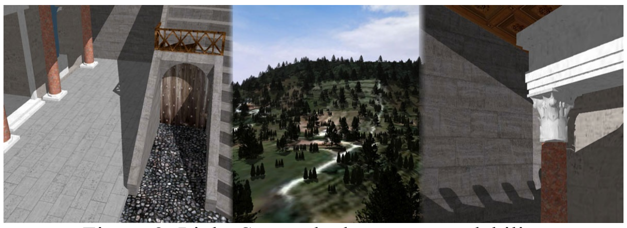


Figure 2. Light-Space shadow maps scalability

A projected texture is used to render the shadow. In this case there is a potential large shadow casting geometry, so the texture resolution should be spread over a large area and this can produce aliasing issues in the shadow. Therefore, the scene has to be spatially well organised to keep the bounds of shadow casting geometry as small as possible. There must be a selection of which nodes will cast shadows and tune the shadow frustum for such large landscapes. The light-space perspective shadow algorithm allows a good and scalable technique from large areas to object details.

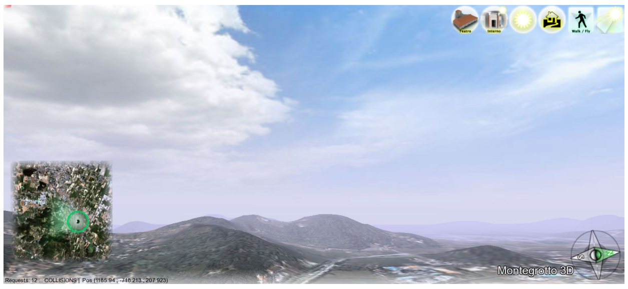


Figure 3. OSG4Web: new overlay 3D interface (icons, minimap, compass, info-area) on Montegrotto Project

A revamped 3D User interface has also been developed, with user customizable modules: compass, mini-map, loading bar, area information and icon bar (Fig. 03).

Every component of the new interface is a 3D object or generic node, and can be loaded from a local resource folder or from a remote location (such as a server) with the purpose of maximizing the customization level from the web page side. A special software layer in fact has been developed up to manage complex interactions: when enabled, it starts listening for commands triggered by user interaction (eg. mouse clicks) directly embedded inside 3D objects. Icons, 3D pins and generic objects are thus able to trigger and activate any defined action or even multiple actions within the virtual world such as flying to defined hotspots, switching models, changing lighting conditions, adding/removing entire 3D models or graphs and other, greatly enhancing flexibility of the application itself.

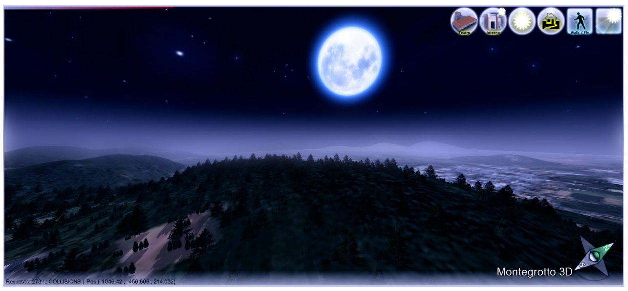


Figure 4. Multiple Sky domes and lighting conditions interactively modifiable (Montegrotto example)

2. A CASE STUDY: RECONSTRUCTING AND INTERACTING WITH MONTEGROTTO THERMAL LANDSCAPE

2.1 Goal of the project

Recently the team has been working into a new case study, focused on the reconstruction of the ancient thermal landscape around Montegrotto town, close to the Euganean hills, south of Padova (north of Italy). The work has been carrying out together with the University of Padova, department of Archaeology. In this case study, an on-line real-time application should be built for two purposes: a public version dedicated to visitor exploration of this territory as it is today and as it was during Roman times, and a restricted version accessible by the team and by all researchers involved in the project. The first application will be fully accessible through the project web site and connected to the web page of Veneto Region, while the second application will be accessed by expert users and used to upload digital contents into the terrain database.

In order to produce a digital library of data useful to be handled over the web a pipeline has been re-defined, based on previous experiences.

Dataset created includes:

- 3d model of the landscape, reconstructed using LIDAR dataset for Digital Elevation Model and Aerial Photographs;
- 3d model of the historical landscape, reconstructed using different imagery, such as historical maps;

- 3d model of the potential Roman landscape reconstructed using artificial geoimagery, based on GIS dataset (in progress);
- 3d models of three archaeological sites, based on Reality-based modelling techniques and acquired using mixed acquisition systems (Laser Scanner, Photo Scan, Computer Vision); (
- 3d models of the three archaeological sites as they potentially have been during Roman times. These models are based on CG-based modelling technique (in progress);
- 3d models of the entire area at lower resolution, based on Procedural modelling technique.
- 3d models of the vegetation at low resolution, based on GIS data.

3. WORKFLOW

3.1 Terrain Generation

3d models of the landscapes have been generated by GIS data. Since the original coordinate system, Gauss Boaga Monte Mario / Italy zone 1, produced coordinates numbers too big (i.e. 1718069.8, 5023118.2), we decided to define a centre of the scene and to create an automatic procedure in the GIS software, GRASS, based on the modification of the Proj library origin. The result is that all information regarding the coordinate system is maintained, while just the origin is moved. All GIS dataset, raster and vector, has been transformed in this way through a script based on GDAL for raster and on OGR for vectors, as in the examples below:

```
gdalwarp -t_srs '+proj=tmerc +lat_0=0 +lon_0=9
+k=0.9996 +x_0=-218069.8 +y_0=-5023118.2
+ellps=intl +units=m +no_defs'
raster_gaussboaga.tif raster_zero_coord.tif
```

Within landscape creation, OSGdem is the provided utility for reading geospatial imagery and digital elevation maps (DEM's) and generating large scale 3D terrain databases that OpenSceneGraph applications can load and browse in real-time. It is part of VirtualPlanetBuilder (or VPB), a terrain database creation tool that is able to read a wide range of geospatial imagery and elevation data and build from small area terrain database to massive whole planet paged databases. These databases can then be uploaded onto the web and provide online GoogleEarth style roaming of whole planet databases, or kept on local disks for high speed access such as required for professional flight simulators. The VirtualPlanetBuilder itself creates databases in native OpenSceneGraph binary format for maximum paging performance. OSGdem has been used to generate 3d landscapes, using in input the zero coordinate DEM and geoimage obtained with the script.

Within the VR Project, content produced with new pipelines has been integrated. Vegetation generators have been used with the goal of populating the virtual scene to enhance the user perceived realism during the navigation experience and the whole Montegrotto Town has been generated procedurally by CityEngine.

3.2 3d modelling

In order to create the Virtual city of Montegrotto two different approach were employed:

- Reality-based modelling, through procedural technique, used to model the existing city and to place in the scene the mesh obtained by scanner laser and computer vision acquisition.
- Computer Graphic-based modelling used to create detailed reconstructions of excavated Roman buildings discovered in the archaeological sites.

3.2.1 Procedural modelling

One of the most important goals to improve the realism of the landscape of Montegrotto and its complexity and to contextualize the archaeological sites studied, was to create a virtual reconstruction of the entire existing city. Modelling a large city with thousands of buildings is always an expensive process which requires huge resources, especially for real time application and web. So, in order to model the virtual city, we decided to use a procedural approach based on grammar and algorithms for automatic generation of complex realities using CityEngine, a procedural buildings generator. This software in fact was developed and optimized for urban and architectural contents and to allow the user a complete control overall entities in the hierarchy of the scene

We started the urban creation importing shape files such as footprints and parcels and using them as allotments for the placement of buildings.

Every allotments is associated with a shape grammar rule which contains a set of rules that define aspect, size and function of each structure.

The biggest challenge was to achieve the whole number and the different typologies of buildings needed to create a plausible urban environment. Using the shape grammar we could define the variables employed to generate different versions of the models and automate the modelling process. For example, to determinate the aspect of the buildings we created a correlation matrix which gives randomly a texture to each building. Into the shape grammar, height information is associated with each texture, so when the software generates buildings, automatically gives them the correct height.

The textures were obtained from a photographic campaign and rectified in order to be adapted to the models. Overall a set of approximately 50 textures was realized.

Using this approach we were able to generate approximately 10000 buildings such as single-family houses, apartment buildings, hotels and churches. These models were very schematic, low poly and particular useful for the real time, in fact their architectural details such as doors, windows, balconies etc., came from texture, not geometry.

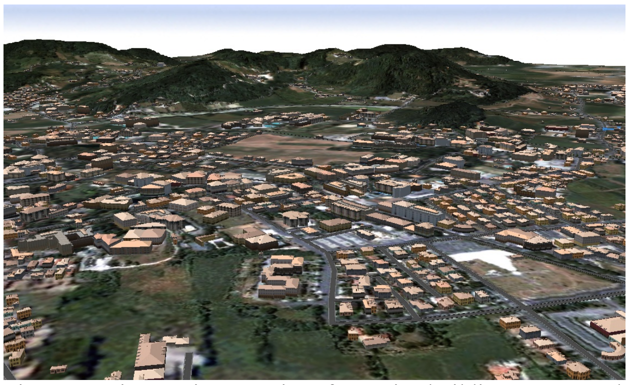


Figure 5. City Engine user interface: city buildings generated with procedural approach

Finally, to represent the archaeological sites we preferred to use detailed mesh acquired in situ with computer vision and laser scanning technique. Using procedural modelling, it was also possible placed in the scene these models, but before the mesh were imported in 3ds Max to decimate the faces. In fact we needed objects with an high level of simplification to be managed also in the real time application.

3.2.2 CG modelling

Hand modelling was applied to reconstruct the main roan architectures discovered in the archaeological sites of Montegrotto, such as the Roman theatre, the baths and the villas. The virtual reconstructions, returning the perception of the buildings in its architectural and decorative completeness, are very useful to a better comprehension of the ancient architectures.

The archaeological 3D reconstructions were realized under the careful guidance of consultants of the Archaeological Superintendency of di Padova and using several types of data as references:

- Floor plans, acquired with a total station
- Laser scanning and Photogrammetry data which document and represent the real topology of the objects in 3D, with a high level detail
- Historical informations
- Archaeological excavation data

Comparisons with similar examples known in the territory of Padova were made to rebuild the parts that had large gaps.

The virtual models were designed with the software Blender and 3ds Max, starting from scans and surveys, to get a complete control of the geometries and graphics details. The Models were created to be used either in applications of computer graphics or in virtual reality environments, therefore, for each of them different levels of detail were created. Textures were created with different techniques depending on the type of surface that we should simulate: when possible, textures were produced by rearranging the images acquired in situ, in other cases, mindful reconstruction and simulation processes of the original surfaces with a digital image processing software were necessary.

Up to now, we have modelled only the architectures of the first century AD, but in the future buildings of later periods could be reconstructed.

3.3 Online World Optimization

The final step is to prepare and optimize all the 3D data for online publication. The terrain has been generated by OSGdem tool, just like in Virtual Rome project, the final result is a well balanced graph, organized in a quad-tree structure that allows tiles (or “pages”) downloading on spatial demand, with Level-of-Detail techniques that provide a multi-resolution representation of the terrain geometry, with the ability to manage very large landscapes thus being suitable in this particular context. (Fig. 06)

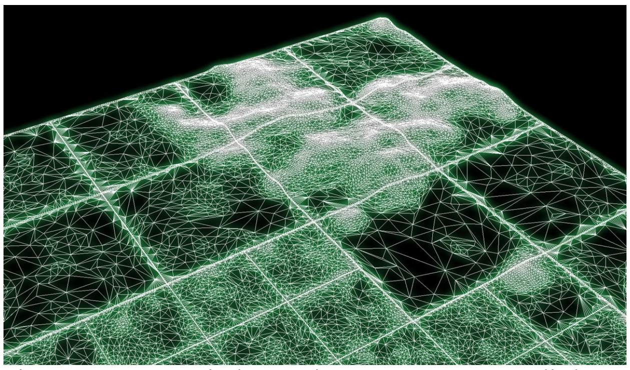


Figure 6. A typical Quad-Tree structure applied to Montegrotto terrain

Level-of-detail technique creates levels of complexity for a given object, and OpenSceneGraph provides methods to automatically choose the appropriate level according to the distance from the user. It decreases the complexity of the object's representation in the 3D landscape, and in this case had unnoticeable quality loss on object's appearance from a distant point of view. Another optimization activated in the visualization engine is to remove on-screen objects smaller than a given amount, removing small features with a further speed-up for example regarding some details of the Montegrotto buildings previously generated by CityEngine. For the Montegrotto city creation, various solutions have been tested. Best results have been achieved developing and using a general purpose “clusterizer” tool, able to subdivide and to organize a very large amount of buildings into tiles and then building a quad-tree hierarchy in a batch process, providing final pages to be uploaded to the server using the web-based Back-End interface. Textures have been optimized and pulled out from optimized geometry, allowing instancing and optimization also during the rendering phase by shared-state manager of the DataBase Pager when clusters are loaded in. (Fig. 07)

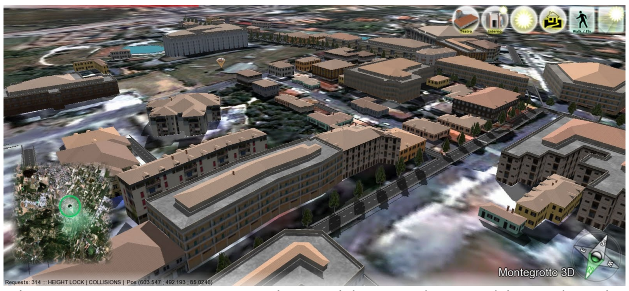


Figure 7. Montegrotto city with quad-tree hierarchy in OSG4Web plugin

The DataBase pager provided by OSG works in several background threads and manages the loading of both static and dynamic database data (paged objects created and added at runtime) being well suited in a web context. The DataBase pager in fact automatically recycles paged nodes outside the current field of view and removes them from the current scene graph when the rendering back-end reaches very high loads, which is when there is a need for supporting multi-threaded paging of massive data, such as the clusters of the generated city.

Being the main focus on some particular reconstructed models scattered across the 3D landscape (such as the Roman theatre) a modular approach has been adopted, with the aim of splitting into smaller components complex given buildings (Fig. modular theatre) distributing the load balance on multiple smaller (paged) files. The modularization also helps the shadow algorithm during the rendering phase for a better spatialisation of nearby objects, removing artefacts due to precision issues.

In this web based scenario, vegetation has been dealt with semi-automated tool “FigiX” used for forest generation. Within general purpose object placing, OpenSceneGraph provides KD-Tree structure (k-dimensional trees). Performance gain with enabled KD-Trees is more than 10 times faster during batch generation of a massive vegetation, organized by the application into clusters below a quad-tree hierarchy that finally generates the output as additional layer for the 3D world. Tree instancing at cluster level is used to tidying up the weight of the given tile and has proven good results in terms of spatial organization and band impact in this context. (Fig. 08)

To maximize performance, redundant and unnecessary state changes have been avoided also at tree level. OpenSceneGraph’s state management helps eliminate redundant state changes and the shared state manager provided by DataBase-Pager is a further speed-up after cluster transmission from the web to the current dynamic scenegraph keeping a light memory footprint.

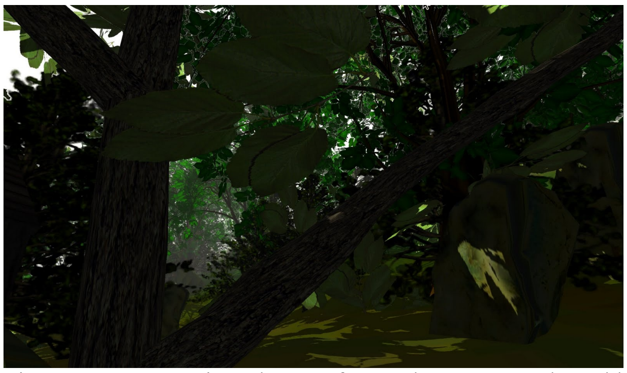


Figure 8. A vegetation cluster of a random generated world online with real time shadows

4. CONCLUSIONS AND FURTHER DEVELOPEMENTS

Regarding future development, addressing complex models publication, either resulting from procedural models, from huge laser scans cloud points or from extensive manual modeling is one of the most challenging problems. automatic model splitting and LOD generation could be applied to leverage on the efficient paging manager provided by OpenSceneGraph framework.

For example, it is likely that an octree paging schema could be applied to point clouds data deriving from laser scan survey, allowing the use of the same web app to document all the steps of the archaeological reconstruction.

Regarding stability, deployment issues and security concern, we found that they need too much development effort, we are moving toward adoption of a general cross platform cross browser framework (Firebreath) for general plug-in development. Having a clear separation between the 3D app code and the Browser embedding framework could help in decreasing the development effort and we are also closely watching the Google Native Client project that aims at providing compile time assessment of native C++ code security.

We are also obviously aware of possible breakthrough changes that the widespread availability of built-in WebGL support inside browsers could possibly change the perspective: browser plug-in have well known deployment disadvantages as they require user to install and trust external developed components..

Currently available WebGL applications still do not seem able to fulfill the performance requirements of landscape sale visualization, but they are fast improving (is Google Earth planning to move on WebGL?) Another factor that could slow the transition to WebGL is the amount of C++ code currently used in scene-graph based apps that has to be rewritten in javascript, this drawback could be mitigate by keeping the javascript app core at minimum while moving as much as possible on the server side; some very early experiment have been carried on with OpenSceneGraph by Cedric Pinson [plobyte].

5. REFERENCES

Barcelò J.A., Forte M., Sanders D.H., 2000 (ed. by). *Virtual* Reality in Archaeology, BAR Internationa Series 843, Oxford

Calori L., Camporesi C., Pescarin S., 2009. Virtual Rome: A FOSS approach to WEB3D. In *3D technologies for the World Wide Web. Proceedings of the 14th International Conference on 3D Web Technology*, Darmstadt, Germany, ISBN: 978-1-60558-432-4, pp. 177-180

Dilla K., Frischer B., Mueller P., Ulmer A., Heagler S., 2009. Rome Reborn 2.0: A case study of virtual city reconstructing using procedural modeling techniques. CAA, Williamsburg 2009, pp 62-66.

Fischer B., 2008. The Rome Reborn Project. How technology is helping us to study history, OPEd, November 10, 2008. University of Virginia.

Forte M., 2005 (ed. by). Virtual Reconstruction of Archaeological Landscapes through Digital Technologies, BAR, International Series 1379, Oxford

Niccolucci F., 2002 (ed. by). Virtual Archaeology. Proceeding of the VAST Euroconference (Arezzo 24-25 November, 2000), Oxford, BAR International Series 1075.

Muller, P., Wonca, P., Haegler, S., Ulmer, A., Van Gool, L. , 2007. Image-Based Procedural Modeling of Facades, *ACM Siggraph*, New York 2007.

Muller, P., Wonca, P., Ulmer, A., Van Gool, L. 2006. Procedural Modeling of Buildings, *ACM Siggraph*, New York 2006, pp. 614-623

Parish, Y., Muller, P., 2001. Procedural Modelling of the Cities, *ACM Siggraph*, New York 2001, pp. 301-308

Pescarin, S., Palombini, A., Vassallo, V., Calori, L., Camporesi, C., Fanini, B., Forte, M., 2009. Virtual Rome. *CAA*, Williamsburg 2009

Pescarin, S., Pietroni, E., Ferdani, D., in print. A Procedural approach to the modelling of urban historical contexts, CAA, Granada 2010

Mozzi, P., Bondesan, A., Busana, M.S., Miola, A., Kirschner, P., Pescarin, Villani, S. M. C. in print. 20,000 years of landscape evolution at Ca' Tron (Venice, Italy): palaeoenviroment, archaeology, VR webGIS. In Van Leusen, P.M., G. Pizziolo & L. Sarti (eds), *Hidden Landscapes of Mediterranean Europe: Cultural and methodological biases in pre- and protohistoric landscape studies.* BAR International Series

Remondino F., El-Hakim S. 2006, Image-Based 3D Modeling, The Photogrammetric Record 21 (115) September 2006, pp. 269-291.

References from websites:
[Firebreath] http://www.firebreath.org

[Google Native] http://code.google.com/p/nativeclient/

[plobyte] http://plopbyte.net/2010/04/webgl-openscenegraph/

[SceneGS] http://scenejs.org/

[WebGL] http://en.wikipedia.org/wiki/WebGL, http://www.khronos.org/webgl/wiki/Main_Page

[X3D] http://www.web3d.org/x3d/content/examples/X3dResources.html#Applications

[X3DOM] http://www.x3dom.org/

5.1 Acknowledgements

We would like to thank the Department of Archaeology of Padova University and Verona University for the cooperation in the 3d reconstruction of Montegrotto VR project, still under development, and specifically Elena Francesca Ghedini and Patrizia Basso. We would like also to thank Veneto Superintendency and Elena Pettenò and the collegue Paolo Salonia, researcher at CNR ITABC.