JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 20 No. 2 (2012), pp. 59-70

Interactive 3D Architectural Visualization with Semantics in Web Browsers

Marcin Książek, Maria Pietruszka

Lodz University of Technology
Institute of Information Technology
Wólczańska 215, 90-924 Poland
mksiazek@stud.ics.p.lodz.pl, maria.pietruszka@p.lodz.pl

Abstract. *This paper focuses on rendering, and access to visual and descriptive information about the digital architectural models on the Web. It was proposed to reach these contents with a help of deep linking, which allows to access to different views and descriptions from both the internal navigation system or from the browser, or search engine. Along with the HTML5 and WebGL it allows updating the link during the exploration of a virtual model, and remembers to re-use. Although all the methods were tested on architecture's models, it can be used in other interactive 3D applications.*
Keywords: *3D visualization, semantic content, WebGL, indexing, deep linking.*

1. Introduction

The main goal of our research is to fulfill the gap between existing reproductions of the architecture (mainly image and the computer animation, sometimes walkthroughs of digital models) and its presentation with the help of new media. Those new media are high quality of 3D interactive graphic compatible with text, picture, animation, video, Web-browsers and search engine. It will reach the 3D model in semantic content, which means additional information about the presented architectonic object or about its constituent's parts. The problems, which have to be solved, regarding:

- storage, transmission and data rendering on the Web,
- coding, extracting and retrieving the semantic content,
- interaction of the user with complicated virtual world.

By 2008 the situation of interactive 3D graphics for the Web has been stable. The VRML/X3D applications were being created, such as THEATRON project (`http://www.kvl.cch.kcl.ac.uk/THEATRON`), and a lot of techniques for describing of 3D objects, their behaviors, and distribution in a virtual space and visualization [1]. These techniques were created from the point of view of interactive 3D application's needs. After the visionary article by Tim Berners-Lee on Semantic Web [2], there was a concept IRVE - Information-Rich Virtual Environments, into which the semantic objects [3]. According to new paradigm of presented 3D media in the WEB, the user should possess also the knowledge connected with the application's purpose and also the knowledge connected with the meaning of the object and its component. This may be knowledge about the object's history, creators, functions, etc. That kind of object is called a semantic one.

In the past few years there have been some interesting ideas for the presentation of the interactive 3D graphics for the Web: the Adobe Flash technology with script language ActionScript 3.0 (2008) and graphics engine called Stage 3D (Fall 2011), the Google O3D (2009), the WebGL technology elaborated by Khronos Consortium (March 2010), and the Microsoft Silverlight 5 (December 2011). Unfortunately, these very young technologies are not stable. An example is the O3D, which was developed by Google for one year only. Therefore, the current is the question S. Ortiz: Is 3D finally ready for the Web? [4].

The second problem is that the tools to encode, search and retrieve semantic 3D media content is not satisfactory yet [5]. Despite this many researchers presented a methodological approach to the semantic description of architectural elements [6], defined an ontology able to describe the shape of 3D model [7] or showed how attribute grammar can be used as a 3D modeling language [8].

In the second (2) section there is presented the particular problem of obtaining information from the virtual worlds and requirements for 3D Web technology are also explained. Then, in the third (3) section a proposal to create IIRVE - Indexed

Information-Rich Virtual Environments, using HTML5 and WebGL is presented. This approach is illustrated as applied to small and large architectural objects.

2. The Problem

Architectural visualization can be classified based on different criteria. Given its purpose, we have: commercial-design visualization, competitions, education and popularization, film, computer games, 3D GIS (Fig. 1), analysis, e.g. selection of optimal illumination, revitalization, and reconstruction (Fig. 2). A. Foni et al. proposed taxonomy of visualization for the application of cultural heritage, taking into account four dimensions: precision (P), interactivity (I), automatism (A), and virtuality (V) [9]. On this basis, they evaluated the sixteen different visualization techniques, from restitution drawings ($P = 0.1, I = 0, A = 0.7, V = 0$), to real-time virtual reality simulations (0.8, 0.9, 0.4, 1), to semantically supplemented 3D (0.8, 0.95, 0.25, 1).

The high level of interactivity of 3D virtual worlds (VR) stems from the fact that the user can freely manipulate and explore it directly. The classical techniques are: free navigation, a list of view points, triggering the behavior of objects and hot spots. An idea of hot spot can be used to provide a context and a meaning associated with the visual content, for example its name, the architect's name, the age of the building, its history, current look (if the building or some part of it still exists). Such content can be both 3D graphics and multimedia content such as text, image, animation, video or audio (Fig. 1). The amount and type of displayed content should depend on the distance of an object or its part to the observer [10], [3].

It is desirable to automatically create semantic content when creating an architectural model. The problem is that the virtual model, enriched with the semantic used not only architects. Depending on the purpose of visualization can also be archeologists, historians, personnel assigned to restructuring or revitalizing activity, curators, teachers, advertisers, investors, etc. Each one adds the information at various stages of creation of the virtual model.

The overall geometrical precision and visual consistency are generally limited by the necessity to provide a real-time feedback to the user. GPU programming and techniques such as normal mapping, light mapping, hemisphere lighting, image-based lighting can help compensate for the lower complexity of the model and increase the realism (Fig. 1 and 2) [11], [12] .

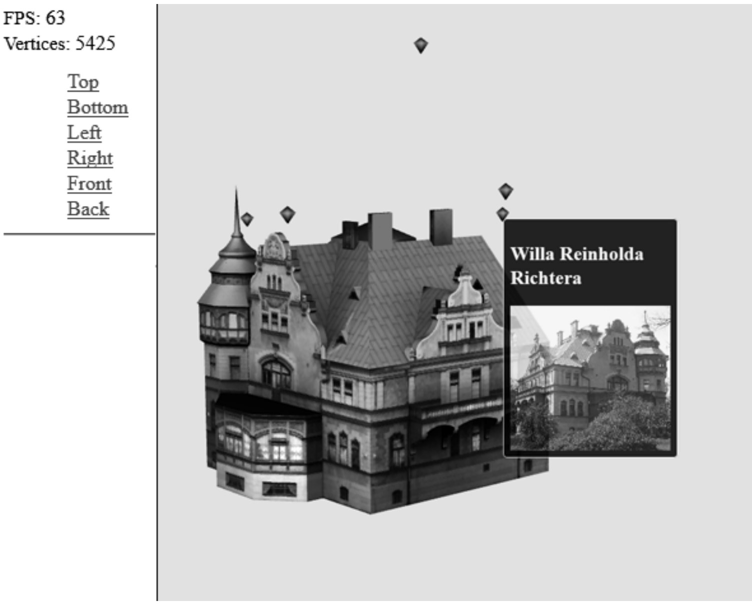

Figure 1. Virtual rector's office of Lodz University of Technology; details of geometry and illumination are stored in textures; clicking on the Right button in the menu or the hot spot, displays the right view of the building, and the text and image associated with it

We were assumed that the IIRVE application should offer different ways to reach the desirable information to the user:

1. Change the view by using the internal navigation system.

2. Selection of the view of the virtual world by clicking on a link on a Web page.

3. Typing the URL address in the web browser.

4. Link from the search results (e.g. Google).

The first one is a typical in the VR systems, the others are specific to the 2D web pages. Unfortunately, the browsers and search engines do not have direct access to "deep" sources of virtual world, run by a plug-in. If the user finds the search view and adds it to the bookmark to find it directly in the future, it turns out that

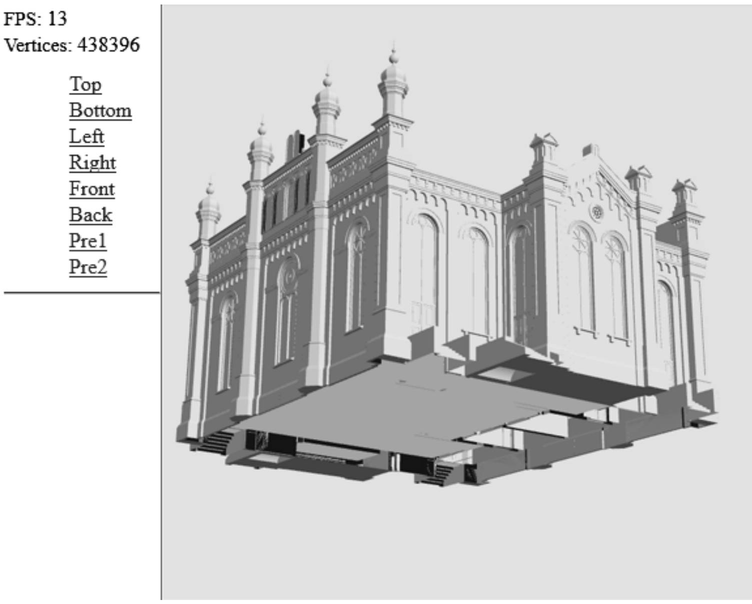

Figure 2. Model of synagogue; only lighting because there is no information about its appearance

the bookmark will contain a link to the default view of VR model, not to the found one! The solution to these problems is deep-linking. This allows assigning the URL address to a particular object or view, located in the content display by the plug-in [10].

Another problem is the choice of 3D technology for 3D Web application. It should ensure the independence of the browser, plug-ins and hardware, as well as support from the graphic libraries (e.g. OpenGL, DirectX). Although any of the currently available technology is not perfect, the most promising are the Flash and WebGL. Both provide a programmable rendering pipeline with Shader Model 2.0 and have several engines and frameworks, such as Away3D for Flash and SceneJS for WebGL.

Flash is stable and popular among designers and consumers. FlashPlayer is installed by a lot more users (99%) than other plug-in-based Web technologies (`www.adobe.com/products/player_census/flashplayer/`).

The WebGL renders the virtual world directly into new HTML5 Canvas element. According to this the users do not need a plug-in to be installed. WebGL works in any browser that supports OpenGL ES (currently Chrome, FireFox, Opera, Safari) and on mobile devices. From the perspective of commercial applications, the disadvantage is the ability to view their source code on user's computer.

3. Proposal solution

The aim of reseach in the field of architectural visualisation is to propose a comprehensive solution from the model to the final images. Such solution was proposed in [10] for Flash. In this section we present the proposition for WebGL taking into account the above requirements and assumptions:

HTML 5 presenting content in web browser,

JavaScript dynamic scripting language for event handling and managing scene and content,

WebGL web standard for a GPU-accelerated 3D graphics API based on OpenGL ES 2.0 exposed through the HTML 5 Canvas element,

SceneJS open-source 3D engine based on WebGL, provides scene graph based on JSON, which is a general-purpose data interchange format,

Collada an interchange file format between modeler and interactive 3D Web application; filename extension .dae,

scenejs-pycollada open-source converter from Collada to JSON, written in Python language,

SketchUp 3D modeler for architects and games developers. The free version can export 3D model to .dae, while the Pro version to .dxf, .obj, .3ds and others.

3.1. Process of creating virtual model

Process of creating virtual 3D environment consists of few steps. First of all acquisition of metric and formal data of the buildings is needed. Next appropriate 3D model have to be created. Popular software used by architects are ArchiCAD, SketchUp and Maya. The first one is able to create both 2D drawings and parametric 3D geometry, also offers BIM (Building Information Modeling) software to the generation and management of a digital representation of physical and functional characteristics of a facility. The SketchUp has been designed with a tagline "3D for Everyone", and offers a parameterization of the model and possibility of being placed in Google Earth.

Maya could be used both to create the model and its editing (segmentation, polygonization, optimizing mesh), texture mapping, and export in DAE format. We applied a render-to-texture techniques for shading (Fig. 1). If there is no photographic documentation of the reconstructed object, and it is important above all, its geometry, the usual technique is render-to-texture ambient occlusion [13]. Instead, we propose the use Hemisphere Lighting shader (Fig. 2) [11], so that there is no need to send the textures over the network.

In addition to the geometry of the building some hotspot elements had to be included. They are responsible for supplying semantic information to the 3D world. Clicking on that point causes appearance of an additional content. Hotspot functionality can be achieved in two different ways. The first idea is to include some additional interactive geometry which can be clicked in order to show some context information. It is simple and intuitive method which does not require any other work with models. Another way is to treat the geometry itself as a hotspot. The object to be presented has to be divided into hierarchical structures. Each element can be bound with additional information. The main drawback of this approach is that hotspot can be unnoticed by user during exploration of the scene. This problem can be partly solved by implementing appropriate appearance of hotspot like changing color of element when mouse is over it.

Before the prepared model can be loaded to 3D world it had to be converted to JSON format which is native scene graph representation for SceneJS engine used in our solution. There are no direct exporters from SketchUp to JSON but there exist two open-source converters from OBJ (obj2SceneJS) and Collada format (scenejs-pycollada). Collada format was chosen because of its flexibility and availability in popular 3D modelers. It supports i.e. hierarchical geometry, advanced materials and custom shaders which are vital for further development of project.

Models created in this way are ready to be loaded and displayed in the Web viewer. However they do not contain any context information. To include some semantic content the JSON scene graph have to be modified by inserting additional "name" node as a parent of geometry:

```
type: 'name',
name: "<h2>Willa Reinholda Richtera</h2>
<img src='image.jpg' width='200'>",
nodes: [
    ...
],
```

“Name” nodes take some HTML-based description as a parameter. This description is being returned when user clicks on the specified geometry.

3.2. Indexing information using IIRVE

Tim Berners-Lee writes: “The essential process in webizing is to take a system which is designed as a closed world, and then ask what happens when it is considered as a part of an open world. Practically, this effect on a computer language is to replace the names/tokens/identifiers for URIs (`www.w3.org/DesignIssues/webize`).

To fulfill requirements imposed by IRVE application deep linking functionalities had to be implemented. When user clicks one of the links the hash part of site’s URL changes and HashChanged event is being raised. This event is being handled by Hash Translator module which translates the hash part of address. Based on its value appropriate parameters of the scene like position, rotation and scale are being set. After this update the render request is being send to the 3D view what causes the appearance of the new state of the scene (Fig. 3).

Also when user interacts with the scene view using mouse or keyboard the URL is automatically being updated. Event coming from Canvas element is being handled by JavaScript code and transferred to Hash Translator. On this step new hash part of URL is being generated and set in the browser window. In the same time scene parameters are being updated and render request send to WebGL context (Fig. 4).

3.3. Results of the test

In performance testing it was necessary to focus on two aspects. First was the rendering speed (frames per second) which had an impact on interaction smoothness (Tab. 1). The second one was time of waiting to load a scene (Tab. 2). All

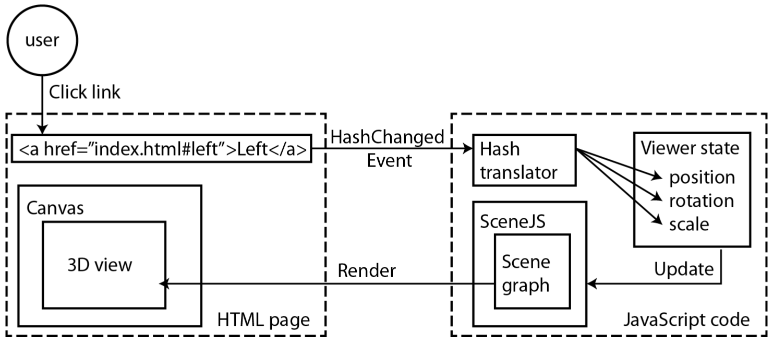

Figure 3. The process of indexing the content in the model IIRVE when user clicks on the link

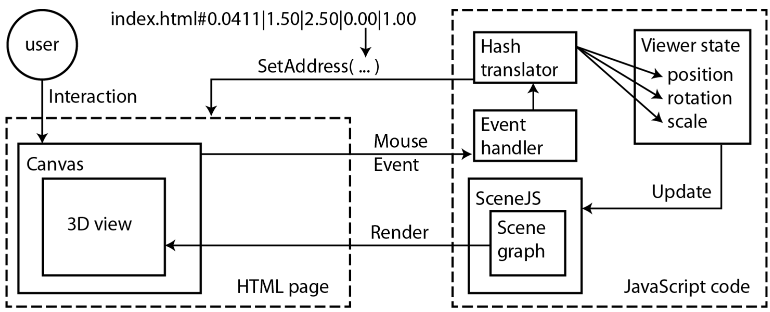

Figure 4. The process of indexing the content in the model IIRVE when user interacts with the scene view using mouse or keyboard

Table 1. Test of rendering speed [FPS]

Scene	**Vertices**	**Firefox [FPS]**	**Chrome [FPS]**
Richter's Villa	5425	63	59
Synagogue	438396	10	15

tests were made for two architectural models on Acer Aspire laptop with Intel Core i5-2430M (2.4GHz) CPU, 4GB DDR3 memory, GeForce GT 540M GPU. The optimal frame rate in browsers was set to 60FPS which is standard in virtual reality systems.

Table 2. Test of internet bandwidth

Scene	**Vertices**	**Data size [MB]**	**Transmitted data [MB]**	**Bandwidth [Mbps]**	**Loading time [s]**
Richter's Villa	5425	1.366	0.838	5 10 54	2.04 1.04 1.09
Synagogue	438396	23.230	2.230	5 10 54	5.45 3.45 1.44

4. Conclusions

The proposed set of technologies, tools and methods can be useful for creating interactive 3D architectural visualization on the Web. As demonstrated by testing a model of average complexity - i.e. about 20 MB - does not cause serious problems. Loading time did not exceed 10 seconds, which J. Nielsen considers the border to discourage user. Lack of loading time speed up when broadband Internet was used results from long waiting time for server response. Rendering speed was dependent on both the algorithm in the graphics engine and the browser (Tab. 1). For future work, we would like to propose a method of automatic creation of semantic content.

Acknowledgments

Authors would like to cordially thank to the architect Rafał Szrajber for providing architectural models.

References

[1] Trumello, E., Santangelo, A., Gentile, A., and Gaglio, S., *A Multimodal Guide for Virtual 3D Models of Cultural Heritage Artifacts*, In: Proceedings of the IEEE International Conference on Complex, Intelligent and Software Intensive Systems, 2008, pp. 735–740.

[2] Berners-Lee, T., Hendler, J., and Lassila, O., *The Semantic Web*, Scientific American, Vol. 284, No. 5, 2001, pp. 34–43.

[3] Bowman, D. A., North, C., Chen, J., Polys, N. F., Pyla, P. S., and Yilmaz, U., *Information-rich virtual environments – theory, tools, and research agenda*, In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, 2003, pp. 81–90.

[4] Ortiz, S. J., *Is 3D Finally Ready for the Web?* IEEE Computer Magazine, Vol. 43, No. 1, 2010, pp. 14–16.

[5] Spagnuolo, M. and Falcidieno, B., *3D Media and the Semantic Web*, IEEE Intelligent Systems, Vol. 24, No. 2, 2009, pp. 90–96.

[6] De Luca, L., Vernon, P., and Florenzano, M., *A generic formalizm for the semantic modeling and representation of architectural elements*, The Visual Computer, Vol. 23, No. 3, 2007, pp. 181–205.

[7] Duan, Y., Cruz, C., and Nicolle, C., *Managing Semantic Knowledge for 3D Architectural Reconstruction of Building Objects*, In: Proceedings of the ACIS International Conference on Software Engineering Research, Management and Applications, 2010, pp. 121–128.

[8] Schmittwilken, J., Dörschlag, D., and Plümer, L., *Attribute Grammar for 3D City Models*, In: Proceedings of Urban Data Management Symposium, 2009, pp. 49–58.

[9] Foni, A. E., Papagiannakis, G., and Magnenat-Thalmann, N., *A Taxonomy of Visualization Strategies for Cultural Heritage Applications*, ACM Journal on Computing and Cultural Heritage, Vol. 3, No. 1, June 2010, pp. 1.1–21.

[10] Andrzejczak, J., Szrajber, R., and Pietruszka, M., *Indexed Information-Rich Virtual Environments: Innovative Solution for Indexing and Searching Content in the 3D Web*, In: IT Tools in Management and Education, Selected Problems, 2011, pp. 179–189.

[11] Rost, R. J. and Licea-Kane, B., *OpenGL Shading Language. Third Edition*, Addison-Wesley, 2010.

[12] Szajerman, D. and Pietruszka, M., *Real-time ice visualisation on the GPU*, Journal of Applied Computer Science, Vol. 16, No. 2, 2008, pp. 95–102.

[13] Apollonio, F., Gaiani, M., and Baldissini, S., *Architectural 3D Modeling for a 3D GIS Web-Based System*, In: Proceedings of International Symposium on Virtual Reality, Archaeology and Cultural Heritage, 2010, pp. 83–86.