Universiti Utara Malaysia
UUM
The Eminent Management University

PROCEEDINGS
iCON C-COMET'14

International Conference on Culture Communication and Multimedia Technology

Sintok, 29 April 2014

Media Transformation

MOHD SOBHI ISHAK
NORSIAH ABD HAMID
AZAHAR KASIM
ZURAIDAH ABU TALIB

SCHOOL OF MULTIMEDIA TECHNOLOGY AND COMMUNICATION (SMMTC), UNIVERSITI UTARA MALAYSIA

Proceedings

INTERNATIONAL CONFERENCE ON CULTURE,
COMMUNICATION AND MULTIMEDIA TECHNOLOGY 2014
[ICON C-COMET'14]

MEDIA TRANSFORMATION

---

29 April 2014
Universiti Utara Malaysia, Sintok, Malaysia

Editors
Mohd Sobhi Ishak, Norsiah Abdul Hamid,
Azahar Kasim, Zuraidah Abu Talib

Media Technology Program
Department of Multimedia Technology
School of Multimedia Technology and Communication
UUM College of Arts and Sciences
Universiti Utara Malaysia
Sintok • 2014

ISBN 978-983-2078-89-0

CetakanPertama, 2014

Perpustakaan Negara Malaysia Cataloguing-in-PublicationData

International Conference on Culture, Communication and Multimedia
Technology (2014 : Kedah)
Proceedings INTERNATONAL CONFERENCE ON CULTURE,
COMMUNICATION AND MULTIMEDIA TECHNOLOGY 2014
[ICON C-COMET' 14] : MEDIA TRANSFORMATION : 29 April 2014
Universiti Utara Malaysia, Sintok, Malaysia /
Editors MohdSobhiIshak, Norsiah Abdul Hamid, AzaharKasim,
Zuraidah Abu Talib.
ISBN 978-983-2078-89-0
1. Mass media--Malaysia--Congresses.
2. Communication--Malaysia--Congresses.
3. Multimedia communications--Malaysia--Congresses.
I. Mohd.SobhiIshak. II. Norsiah Abdul Hamid. III. AzaharKasim.
IV. Zuraidah Abu Talib, V. Universiti Utara Malaysia,PusatPengajian
Teknologi Multimedia danKomunikasi. VI. Title.
302.2309595

TABLE OF CONTENTS

SUB-THEME 3: BROADCASTING

SUB-THEME 4: COMMUNICATION AND NEW MEDIA

SUB-THEME 5: MULTIMEDIA, INFORMATION SYSTEM AND INFORMATICS ENGINEERING

Universal Framework for Displaying Multi-format Web3D Worlds

Mursid W. Hananto
Department of Information System,
Universitas Ahmad Dahlan, Yogyakarta
E-mail: toxynusca@yahoo.com

ABSTRACT

There are various format for displaying 3D graphics on the web. Although there has been a standard format defined by Web3D consortium, it is known that there are a variety of format and incompatible standards still in use and continue to be developed. This raises the issue where not all web browser software can display Web3D world and its 3D objects, even if the browser has met certain requirements, or have been used the required plug-in/add-on. There is a need for solution so that users can display a variety of formats easier and efficiently. This work in progress is proposing a universal framework as an attempt to solve the problems stated above. This framework will not only be a plugin or tool, but it will be an integration of various components, including the core of many Web3D format currently exists. The proposed framework will used a format as the main base, where all other formats that are recognized by the framework will be converted into the main format. Implementation is expected in the form of a specific browser for displaying Web3D graphics and also be able to work with conventional browsers to display either the Web3D worlds or conventional web pages.

Keywords: Web3D, multi-format, framework, universal, browser.

INTRODUCTION

Web3D in general is a platform to display 3D graphics on the Internet by the means of web browser. Unlike conventional web which is fully two-dimensional (2D), Web3D is using a different basic display pattern in the form of three-dimensional (3D) graphics which applied to sites that are displayed in a web browser. The particular display pattern allows users to browse the web not in a conventional way, but as in the real world. Browsing through web content is no longer inter-page and inside pages, but turned into a "movement" for the user as if engaged in a site he visited. This movement is done as a form of navigation inside the "world" which is an implementation of 6 degrees of freedom in motion just like they do in the real world. Nevertheless, Web3D has been prepared from the beginning to complete the conventional web. It is not to replace it because it allows many features that are difficult or not simply be given by conventional web (Pesce, 1995). As such, some things that are more appropriate by conventional web display (such as text) will be better using 2D principles although Web3D can also display it.

There are several facts related to Web3D. After VRML was shown for the first time in 1994 by Mark Pesce and Tony Parisi in a conference organized by World-wide Web Consortium (W3C), VRML then set into a Web3D standard format with standard number ISO/IEC 14772-1:1997 (Carey et al, 1997). Furthermore, the 2002 amendment adds geospatial features and NURBS in this standard so that it becomes more refined (Web3D Consortium, 2003). In addition to VRML, Web3D has another standard format which is X3D that emerged in 2004 with standard number ISO/IEC 19775-1:2004 (Web3D Consortium, 2004), and there have been several amendments made for this format, including last amendment of 2006 to further refine it (Web3D Consortium, 2011).

A Web3D world displayed in a browser can have a simple or complex shapes, and can also be constructed in a distributed manner by components from a variety of sources across the world. Web3D documents are stored in ASCII type files so that it can be opened with a text editor. A complex world will be difficult to make with just relying on a text editor, so it can utilize the help of a general 3D object scenery development software, and then converted into the appropriate format which comply to Web3D standards. Web3D also requires adequate graphics subsystem in a computer to visualize the 3D graphics in the browser, something which has now become a common capability for computers manufactured in 2010 and thereafter.

There are many formats out of Web3D standards, including WebGL, JOGL, Java3D, Flash3D, O3D, Cult3D, Anfy3D, and others. Since they are not conformant to Web3D standards, some formats require specific tools to create a 3D world, some may use 3D object creation tools that are commonly found even though they are limited in functionalities. In addition, some proprietary formats require additional costs for both the manufacturing tools and to distribute the products. Nevertheless, there are also Web3D format that do not require any additional software for the browser, but with certain requirements that must be met. With so many formats, it is not possible to force users to use a particular format, although it has been referred to as the standard format.

The problem that arises is that all of these formats are not compatible with each other. This problem can also be seen between VRML and X3D which are Web3D standards. Both are not compatible even though X3D code can be written by means of writing VRML. Some formats require certain plug-ins while others do not, but those which do not need plug-ins still require some specific prerequisites that have to be met so that the browser can display the 3D world of that particular format. Some formats that do not require plug-ins include WebGL, X3DOM, and newer version of O3D, as well as formats that use Java. Newer non plug-in formats which have been able to utilize the acceleration of 3D graphics adapter on the computer (except those using Java) generally requires that the browser should be fully compatible with HTML5 and optimized JavaScript, and OpenGL libraries is available. The browser must also fully support the HTML5 Canvas element so that JavaScript can draw 3D graphics directly on the page displayed in the browser. Not all browsers support this, in a particular browser this requirements have to be set up in the form of a browser plug-in so that the browser can display a format which should not require plug-ins.

EXISTING IMPLEMENTATIONS

In the early development of Web3D, there is no browser that can display 3D graphics in the web. A specific Web3D browser was made at the beginning of Web3D introduction, a browser that allows simple navigation on 3D objects which is a banana model (Pesce, 1995). This first experimental Web3D browser version was made prior Web3D standard formats which is VRML 1.0 was set.

Web3D browser advancements are still in motion primarily due to the needs to keep pace with conventional web technologies. There are Web3D specific browser built using Java or C + +, both for standard format (Blais et. Al, 2001) as well as non-standard or proprietary format which uses its own protocol. Standards and Web3D technologies continue to be developed and include a variety of technologies, which has been used extensively (Walsh, 2003), thus increasing the ability of the browser. Nevertheless, the standard format browser can only display Web3D standard formats, and this also goes to proprietary formats. Non-standard formats usually have provided a special browser or formed into a plug-in.

As a form of visualization that is different from the conventional Web, Web3D objects generally appear in a different window than conventional web view. It can be found either in the standard format and non-standard formats. This is because objects in Web3D have different visualization method compared to conventional information on the web. Several ways have been used in order to make Web3D visualization appear integrated with conventional web pages, including the use of a frame containing a world which is placed in a conventional web page (Beard, 2006). Even though the page appears containing a 3D object, both are actually separate the information display. Later, Web3D standards have set up a mechanism for integrating information, using internal scripts and coding performed on the URL (Stepan, 2003). Efforts to integrate conventional and 3D information also performed using the push-pull-process model (Thorne, 2005) to make it possible to use a conventional web service page to manipulate a 3D world and vice versa.

Web3D information integration efforts are generally made to unify the information in conventional form with Web3D world view. The integration of information between 3D objects in different formats can also be done on a web page, using inline frames provided with necessary link to a URL. This method allows multiple Web3D formats to be simultaneously appear in a conventional web pages. But this method also requires that the browser must have the ability to display all the formats. Since each frame is invoked via the URL, if the URL can not be reached then the world which becomes the frame contents will not be able to be displayed. While it can display many formats at once, basically all existing world view is not fused and can not be directly connected to each other. Despite being in one page, Web3D formats which are not mutually compatible with each other can not be interconnected directly and the content integration does not occur.

Because there are numbers of formats in Web3D, prospective Web3D site developers are quite difficult to make a choice for their needs, so it needs to be done comparatively. Comparative study between Web3D formats has been undertaken by Turonova (2009) in order to find a format that is considered most appropriate to be implemented in 3D Mobile Internet project. Based on these studies, X3D which is a standard format was recommended. This was because the standard format is stable and has a lot of features, and it is quite popular. Another study has been conducted by Waerner (2012) which aimed to review some 3D graphics technology in the web that can be utilized in a web-based application. The research has chosen one of the Web3D technologies that is WebGL as the primary format for the resulting prototype.

The makers of popular browser today have some direction in creating browsers that can accommodate as many Web3D formats as possible. Some makers including Google initially made the plug-in because of the need to access the API in order to deliver better performance on a complex world (Paul, 2009). O3D proposed by Google is different from the Web3D consortium standard format. Nevertheless Google itself did not pursue this format and tend to support WebGL (Shankland, 2010). Some other makers are trying to agree on a browser that support 3D graphics on the web wherever possible that does not require plug-ins or add-on (Takahashi, 2011). Browser makers who want their products to be plug-in free are making their efforts so that 3D graphics can be hardware accelerated in HTML5 compatible browsers, using WebGL format that are able to take advantage of OpenGL commands via JavaScript (Burns, 2011).

INITIAL ATTEMPTS TO SOLVE THE PROBLEMS

On the developer side, they can choose any format of the current choices. This creates difficulties for the user to adjust to the format used to describe the world. Initial attempts to solve the problem that has been done by users are:

1. Installing as many plug-in as possible.
   Each plug-in only works to display a particular document format. The plug-in may incur compatibility issues with browser, and may have a potential problem with other programs/applications installed in the computers.
2. Using browser which complies to as many compatibility requirements as possible.
   Standard browsers have been designed to comply with W3C. In reality, not all browsers are equal in their abilities/features, and browser might display pages which are not always have the same look as when it's displayed on other browsers. Even this way can not be used when the format to be displayed needs plug-in.
3. Converting different object/world into one format.
   Although it appears to be a good solution, but not all format have conversion tools available. It will also add to the hassles since users have to search for appropriate conversion tools. Users still need to find an object or world to be converted, and must specify the destination format. The result of such a conversion is not always desirable, especially if there are any need for adjustments.

Of all variety of efforts, the results obtained are different. Plug-ins can indeed be a simple solution but with a few notes. Web3D plug-ins that are currently available have not been able to display more than one format. A format can also have more than one different plug-in because it is made by a different manufacturer. Plug-ins can have the potential to deliver a problem to the system when there is a conflict with another program that is installed on the same computer. Although based on the same world document format, the capabilities and features of each plug-in is not always the same. Different plug-in also provides different interface and interaction to users even though they display world with the same format.

When the preferred solution is using a browser that meets as many compatibility requirements as possible, only a certain format are viewable. If the browser fails or has a problem to display a world, user can use other browsers. In terms of practicality, this is not a good choice since it adds more task to user before they can see the world in a browser.

Better results obtained if the plug-in has been added to the browser. If the format is unsupported by the plug-in, user can attempt to do the third way: converting documents into formats supported by browser or plug-in. The combination of these ways will make it difficult for general users who want the ease of viewing variety of Web3D formats.

PRIMARY REQUIREMENTS

Based on the problems mentioned above, as well as some things that can be noted from the early attempts to solve the problem, the primary requirements needed are the ability to read, to convert, and to display a variety of world format in real-time, using a tool that does not force the user to do a lot of extra work. Basic features for this solution can be obtained from the use of plug-ins, along with browser that has high compatibility, and an ability to convert the format if necessary. This capability is then integrated in the form of a universal framework that makes it possible to display a variety of existing Web3D formats.

Although it has the ability to display the Web3D world in various formats (the ones recognized and registered to the framework), this framework does not specifically look at the scale, position, and orientation which are internal attributes of the object. This is because it requires advanced scripting ability that leads to a platform or a new format in order to modify objects on-the-fly.

PROPOSED SOLUTION

Based on the problems and requirements stated above, a solution is proposed in the form of a framework that is expected to be compatible with a wide range of Web3D formats. The framework will bring together information and the ability to display and interact with current existing format. The proposed framework will consist of several components which can be seen in Figure 1. The framework of this proposal is not specifically intended to handle any unrecognized format, the feature will be part of the next works. The framework has components as follows:

1. Set of reference format.
   This component works like a plug-in to the browser, where additional formats inserted into the framework as a library. If there are documents formatted in accordance with the format of the reference list (a list of compatible formats), it will be converted to the main format.
2. The main reference format.
   The main reference format is the actual format that is used as the base document to be displayed to the screen. It is hoped that by using only one format, then the system will be consistent in its performance. This component will be prepared to be able to handle a world with multiple formats. All the different formats are to be converted to the main format before being displayed in browser. This method is expected to minimize the potential conflicts that may occur when different format of objects rendered to the screen together.

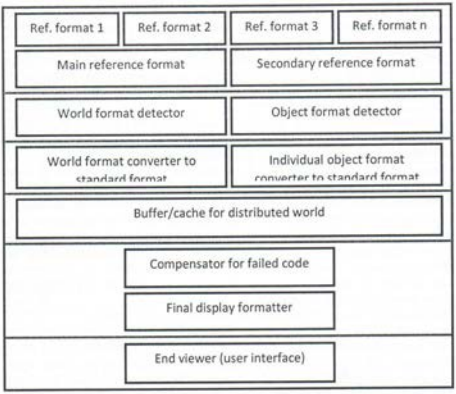

Figure 1: Proposed Framework Components

3. Secondary reference format.
   In addition to the main reference format, there is also a secondary reference format that can take over the role of the main reference when some of the world or a particular object is difficult to translate to the main reference format.
4. Format detector.
   The detector is used to determine the document format used before being modified to reference format. In a distributed world or large world consists of a collection of smaller world, the detector will search for whatever format contained in the whole world elements.
5. Format converter.
   Format converter will convert the format of Web3D documents to be displayed when the world does not conform to the main reference format. Conversion will also be done to world elements of distributed world into reference format used in the framework.
6. Buffer/cache for distributed world.
   Distributed world does not always look perfect when there is a world element having trouble displayed, which are due to problems in the data transmission path. Buffer or cache is used to store all the elements of a distributed world so that the complete world can be displayed entirely.
7. Compensator for failed code
   Elements of the world that can not be converted to reference format can make the entire world failed to be displayed. The compensator will attempt to resolve this by generating an object in the area that is supposed to contain the world elements. This component also handles error so that all the other parts of the world can be displayed in the browser.
8. Final display formatter
   The display formatter is basically a component that will be used to enforce a uniform pattern of overall world view. This component will also apply all the effects that have been set in the primary world document to the entire world elements.
9. End viewer/browser
   This components is the main interface to users. In this section, the final result of the whole process will be shown to user. User interaction to the world is also performed through this section.

The proposed framework outline processes are shown in figure 2, the explanation is as follows.

1. Documents containing the world are entered into the framework.
2. The framework will attempt to find out whether the world format is in accordance with the main format in the framework.
3. If it complies with main format reference, it will be taken straight to the end viewer.
4. If the format does not comply with the main format, it will be compared to the library to see if it is in accordance with the additional format.
5. If the world format is in correspondence with additional reference format in the framework library, it will be converted to the main format.
6. After conversion process has been done, adjustments to size, scale, and orientation of the entire world will take place.
7. The results of the adjustment will be taken to the end viewer.
8. If the world does not match with the format of the existing reference in the framework, then the framework will display a message that the format is not recognized. The framework will bring up a place-holder in the form of an object marker used to indicate that the object which should appear on the area can not be displayed.
9. If more than one world document will be displayed at once, the framework will look at each format used by the worlds. If there is an object or a world that does not match

with the main format, it will be compared to the reference format library contained in the framework. The next step will then be carried out as in step 5 above.

Generally, implementation of the proposed framework can be realized in the form of specific Web3D browser, not in the form of plug-ins. The plug-in option is not selected because of the possibility of incompatibility with existing or future browser. This potential problems which quite considerable but difficult to estimate may reduce the overall performance of the browser. Further research is needed to determine the effectiveness, capability, as well as a variety of other factors that can affect its performance. The implementation also does not lead to the creation of new standards.

Proposed specific Web3D browser should be able to accommodate plug-in formats that may not be handled by this browser, and in the next generation will allow the addition of the new format module in order to expand its native compatibility with a wide range of formats. Although intended as a specific Web3D browser, the browser must also compatible with HTML 5, JavaScript, and Java to make sure that the browser can display formats using those particular language.

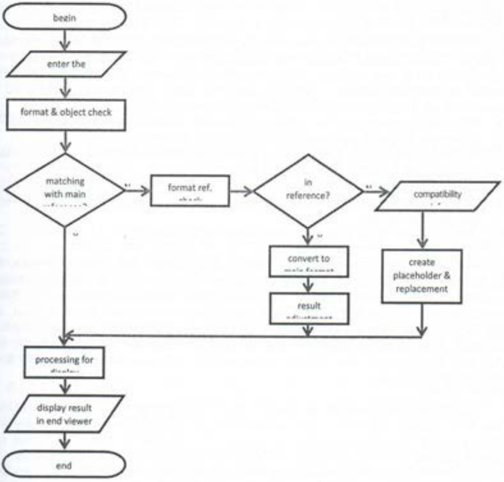

Figure 2: Processes in Proposed Framework

The Web3D browser should be able to access the graphics adapter natively through the use of DirectX and OpenGL APIs. Support of graphics adapter and with 3D acceleration will obviously be very beneficial to the performance of a Web3D world. Graphics adapter with 3D acceleration features will speed up the display so as to provide an enjoyable experience for site visitors, even in a complex world.

Although HTML 5 standard has the addition of Canvas element that allows for the display of graphics objects directly in the browser, HTML 5 can not directly access the API that allows the utilization of accelerated graphics adapter. In order to display 3D objects, JavaScript is required. 3D graphics API access via scripts which then have to be translated by browser will provide a performance that is not as high as when directly accessing a hardware that can provide 3D acceleration. Therefore this method is not selected as the primary format in the implementation of the proposed framework.

Therefore, the existing standard formats and has been widely used will be proposed as the main base forms in the proposed framework. WebGL as a non plug-in format can be used as the basis of the secondary format because it has a prospective wide usage. Although WebGL is not a standard, some of the popular browsers have shown their support for this format.

Problem that still exists is when there is a distributed world which contains more than one format. Initial proposal is that the framework will have detection capability to the world/object to be displayed. If there are more than one format, then all other format will be converted to the main format of the proposed framework, before being displayed to the browser. This will bring up the overhead time, it can occur due to:

1. detecting too many objects or there are complex objects contained in a world;
2. determine that particular object complies with main reference format;
3. comparing the object against stored compatible reference format when a world/object that does not match the primary format is encountered;
4. converting the objects if needed;
5. displaying to the browser according to the appropriate size, orientation, position, and the intended scale;
6. applying visual effects set on the main part of the world.

A separate study is required to review and optimize the core processes. Further study needs to be carried out on the conversion of one format to another format, their complexity, process performance, and how to handle the conversion which not producing the expected results. At this time, distributed Web3D world is known to be existed, but the worlds forming must have the same format. Distributed world with different world formats has never been found since the browser must be able to recognize the various types of world format. The proposed framework does not specifically address these issues, but it can be directed for this purpose.

Based on the proposed framework, the first step that needs to be done in future studies is to choose a format which will became the primary format of the framework, all other format will be converted to this main format. A new or standard format may be proposed, but this option is not taken since it will take more time to develop the new format. To support this effort, comparation study of characteristics and many other aspects (e.g. ease of use, popularity, etc.) will have to be done to obtain initial information from the format required.

CONCLUSION

The proposed framework is a general working step along with its forming components. It includes several key features including the ability to detect a particular Web3D document format. The basic principle used is to use a format that will be the main base format to display the results in a browser. All format has to be known by the proposed framework so that it will display the expected results.

Proposed framework is expected to be a solution that became an easy and efficient way in the effort to display a variety of 3D graphics formats in the web. Research that is specific to each section proposed to be an actual implementation of the framework are still needed in order to support the development of a solution for displaying a variety of formats Web3D.

FUTURE WORKS

The current work is still in progress. Further research is needed to detailing the proposals presented so that potential framework will be further investigated in accordance with the need to display a range of Web3D formats simultaneously in a single browser. Future studies will look at every part of the proposal as well as the changes and adjustments that must be made in order to redirect to the actual development of the necessary components in the overall framework.

REFERENCES

Beard, D. (2006). Using VRML to Share Large Volumes of Complex 3D Geoscientific Information via the Web. In Proceedings of *Eleventh International Conference on 3D Web Technology* (pp 163-167). New York, USA: Association for Computing Machinery. doi: 10.1145/1122591.1122614.

Blais, C., Brutzman, D., Horner, D., Nicklaus, S. (2001). Web-based 3D Technology for Scenario Authoring and Visualisation: the *SAVAGE* Project. In Proceedings of *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2001.* Retrieved from www.dtic.mil/dtic/tr/fulltext/ u2/a422086.pdf. Orlando, Florida, USA: National Training and Simulation Association (NTSA).

Burns, M. (2011). *Plug-in free 3D graphics for HTML5 Web browsers.* Retrieved from http://www.macworld.com/article/1158363/webgl.html. San Francisco, CA, USA: IDG Consumer & SMB.

Carey, R., Bell, G., Marrin, C (1997). *ISO/IEC 14772-1:1997 Virtual Reality Modeling Language (VRML97).* Retrieved from http://www.vrml.org/ Specifications/VRML97. The VRML Consortium Incorporated.

Paul, R. (2009). *Google joins effort for 3D Web standard with new plugin, API.* Retrieved from http://arstechnica.com/information-technology/2009/04/ google-releases-3d-graphics-plugin-for-browsers/. California, USA: Conde Nast.

Pesce, M. (1995). *VRML: Browsing & Building Cyberspace.* Indianapolis, USA: New Riders Publishing.

Shankland, S. (2010). *Google scraps plug-in, refashions 3D Web plan*. Retrieved from http://news.cnet.com/8301-30685_3-20004490-264.html. San Francisco, CA, USA: CBS Interactive Inc.

Stepan, V. (2003). *Using VRML EAI in Applets and Applications*. Retrieved from http://mira.svalbard.cz/pubs/Using_VRML_EAI_v2.pdf. Prague, Czech: Vladimír "Míra" Štěpán.

Takahashi, D. (2011). *Browser makers release standard to bring 3D graphics to web apps without plug-ins*. Retrieved from http://venturebeat.com/2011/03/03/browser-makers-release-standard-to-bring-3d-graphics-to-web-apps-without-plug-ins/. San Francisco, CA, USA: VentureBeat.

Thorne, C. (2005). Exploiting an Evolutionary Accident in Web3D Communications to Integrate Application Components. In *SIGGRAPH '05 ACM Web Program* conference proceeding, Article No. 1. New York, USA: Association for Computing Machinery. doi: 10.1145/1187335.1187337.

Turonova, B. (2009). *3D Web Technologies And Their Usability for The Project 3D Mobile Internet*. Retrieved from http://www.rdc.cz/download/publications/Turonova_2009_3DWeb.pdf. Research and Development Center for Mobile Applications, Faculty of Electrical Engineering, Czech Technical University, Prague.

Waerner, M. (2012). *3D Graphics Technologies for Web Applications*. Master Thesis in Information Coding, Department of Electrical Engineering, Linkoping Institute of Technology, Linkoping, Sweden. Retrieved from http://liu.diva-portal.org/smash/get/diva2:536657/FULLTEXT01.pdf.

Walsh, A.E. (2003). Emerging Web3D Web Standards and Technologies. In *6th International Conference on Humans and Computers*. Retrieved from http://mediagrid.org/publications/papers/Emerging_Web3D_Web_Standards_and_Technologies.pdf. Fukushima, Japan: University of Aizu.

Web3D Consortium (2003). *VRML97 Functional Specification and VRML97 External Authoring Interface (EAI)*. Retrieved from http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/. Web3D Consortium.

Web3D Consortium (2004). *ISO/IEC 19775:2004 — Extensible 3D (X3D)*. Retrieved from http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/. Web3D Consortium.

Web3D Consortium (2011). *X3D Architecture and base components - Amendment 1: V3.1*. Retrieved from http://www.web3d.org/realtime-3d/specification/19775-1/v31/x3d-architecture-and-base-components-amendment-1-additional-functionality. Web3D Consortium.