Scene integration for online VR advertising clouds

Michael Kalochristianakis, Markos Zampoglou, Kostas Kontakis, Kostas Kapetanakis, Athanasios Malamos

Department of Informatics Engineering, Technological Educational Institution, Heraklion, Crete

***Abstract* — This paper presents a scene composition approach that allows the combinational use of standard three dimensional objects, called models, in order to create X3D scenes. The module is an integral part of a broader design aiming to construct large scale online advertising infrastructures that rely on virtual reality technologies. The architecture addresses a number of problems regarding remote rendering for low end devices and last but not least, the provision of scene composition and integration. Since viewers do not keep information regarding individual input models or scenes, composition requires the consideration of mechanisms that add state to viewing technologies. In terms of this work we extended a well-known, open source X3D authoring tool.**

***Keywords* — Virtual reality, web advertising, 3D technologies**

I. INTRODUCTION

ADVERTISING through the world-wide web has been gaining attention during the last decade in order to cover the needs of enterprises for promotion via the cyberspace. As 3D technologies become more ubiquitous, virtual reality (VR) advertising becomes an appealing possibility. VR allows full control over digital worlds that is, the potential for creativity without limits. Although this type of creativity has boosted fields relating to multimedia it is still rather immature as far as marketing is concerned. There has been research on the field such as evaluations that designate the advantages of enabling VR technologies. For instance, the impact of allowing users to interact with the virtual representation of products is significant both in terms of appeal, penetration and general product awareness [1] [2] [3]. One of the issues regarding the general application of VR approaches in advertising is cost. 3D content is in general more expensive to produce in comparison to more traditional multimedia such as video, audio, images and text. The design of models and composite scenes, visual effects and interaction inevitably requires technical knowledge and work that cannot be allocated in terms of projects with narrow scope, besides artistic design and work. Productive commercial VR solutions are typically produced upon platforms or engines.

Our approach in the field of VR in terms of the iPromotion project aims is to provide an online infrastructure that will allow advertisers to market products and services by building and exposing experiential environments for their end users. They will be able to include ads within web page banners or as web page pop-ups in terms of traditional online marketing but they would also be capable to produce immersive experiences through other interfaces, such as interactive widescreens or touch tables. Our system is design to support both the composition and the exposure of VR content by means of cloud-enabled storage and services. The advantages the new advertising paradigms are obvious; advertisers can build and distribute their content using central, online infrastructures. Through appropriately designed 3D worlds advertising can provide a direct representation of the ideas, messages or product characteristics based on exciting interactions that may take any wanted form ranging from multimedia stimuli to games, puzzles and interaction scenarios. At the same time, users or potential customers will be given the opportunity to take part in a simulated experience regarding the advertised goods.

In order to sustain the aforementioned scenario it is essential to provide users with the means to compose any desired virtual worlds that is, not only to provide them with sufficient amounts of VR content units that is, 3D models or scenes, and also with the means to the content for the composition of complex scenes or even worlds. Complying with established standards is a means to accomplish only the former; even if the provision of significant amount of content units can be based on offering standard, royalty-free or available complying content for our platform, the ability to build more complex content cannot not supported by any. The scene composition module of our platform is designed to provide this use case by adding visual controls to content units so that they can be managed in terms of broader environments. It takes advantage of the appropriate event models to store information about the position, size, scale and rotation at least, in order to create simple representations for complex worlds so that content units are reused by accessing the cloud of our platform [9].

The rest of the paper is organized as follows: Section II presents the related work in the field of 3D scene representation, the standards and the tools. Section III briefly elaborates on the research project that produced this work, the iPromotion VR platform. The section overviews the architecture and components of iPromotion, focusing on the aspects that offer contributions to the VR field. Section IV

DOI: 10.9781/ijimai.2014.312

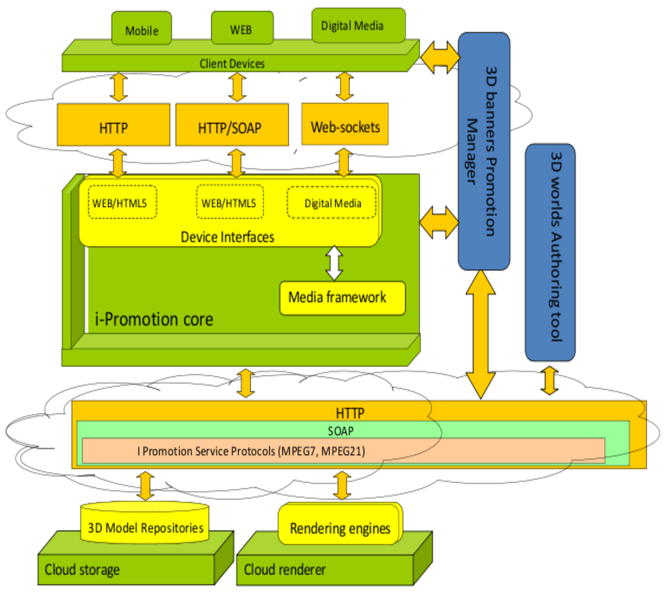


Fig. 1: the component architecture of the system

is the primary topic of this paper. Finally, section V discusses our conclusions and explores potential next steps in the development our scene composition software.

II. Procedure for Paper Submission

A. 3D technologies

The evolution of cross-platform web applications has triggered the use of XML-encoded textual files that provided a platform-independent solution for information exchange. In the field of two dimensional graphics, Scalable Vector Graphics (SVG) is an XML-based open standard that provides the necessary tools for development. SVG is capable to produce accurate results by describing vectors in XML tags. Graphics can thus be re-sized or zoomed without any loss of quality and can also be transferred in the same manner as any other XML-encoded textual files. The protocol can also utilize text annotations withing the images in order to provide supplemental functionality. Search engines or end-users can search such text annotations to retrieve content based on search criteria or individual needs. SVG is also a W3C recommendation [6] and can thus be integrated the Document Object Model (DOM) and eXtensible Stylesheet Language (XSL) besides other standards. Thus, interactivity and animation can be applied in SVG representations through object module manipulation languages such as Javascript.

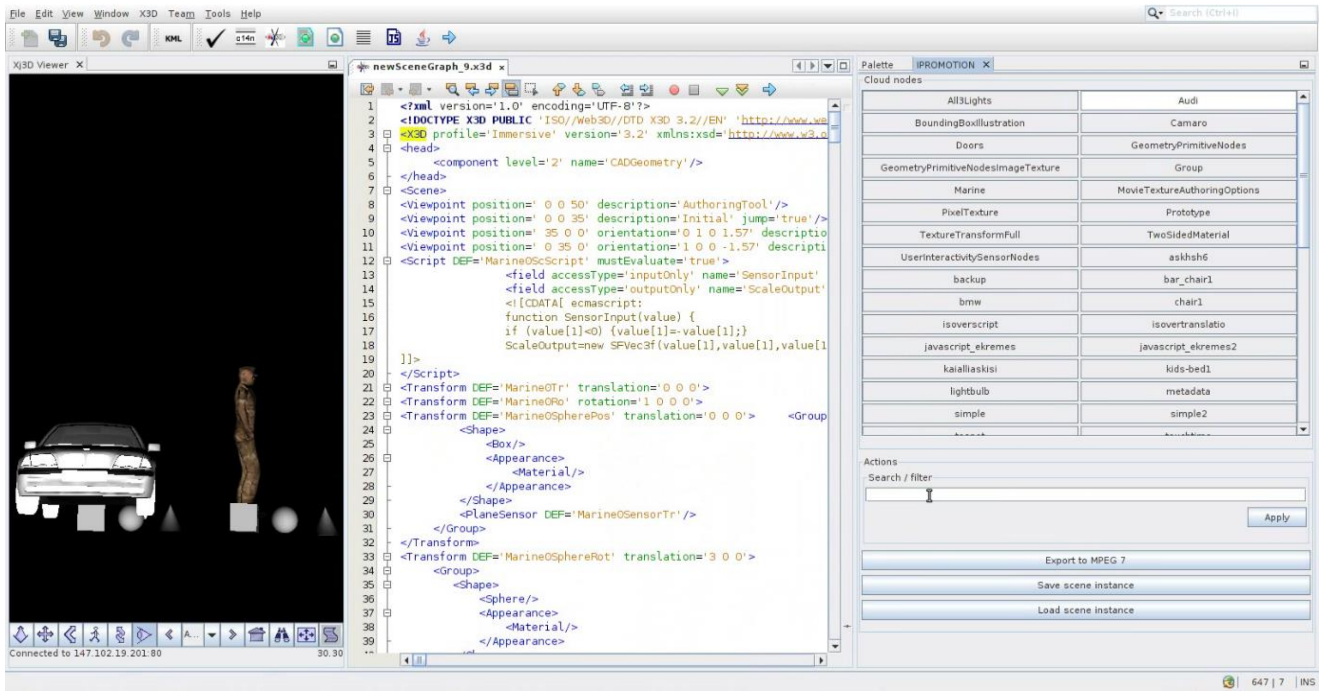
X3D is a royalty free, open, ISO ratified standard for developing 3D graphics and also a run-time architecture to represent and communicate. X3D is developed by the Web3D Consortium [7] as the successor of the Virtual Reality Modeling Language (VRML). In X3D, representations use XML-encoded format and thus information is platform independent and ideal for use over the internet. X3D supports real-time communication and integration with web-service architectures, distributed networks, cross-platform applications, inter-application information transfer. It is modular by nature and is thus capable to support the concatenation or superposition of 3D components. The standard is capable to support, besides high-quality graphics, real-time interaction and audio-video sequences. The functionality of the standard is structured in layers called profiles that define the functionality subset that is to be used. The run time environment of X3D is essentially a browser engine that carries the rendering logic. The engine is released as a standalone application, as a web browser plug-in and as an open development project.

B. Scene authoring

The need to develop and combine X3D models and scenes within our platform led us to consider the available tools and their potential to be extended, maintained and used. The relevant technologies include X3D players, plug-ins, development environments and authoring platforms. A cohesive overview can be found in the pages of the WEB3D consortium. There is also an interesting comparison of the authoring and scene conversion tools in the paragraph that regards authoring. The table presented there compares most of the relevant tools that is BS Editor, SwirlX3D, X3D-Edit, Flux Studio and Vivaty Studio besides some conversion tools. The information is kept updated since it refers to the current versions of the software. One can observe a number of points. To start with, full profile is not completely supported by any authoring tool while it is “nearly supported” by BS Editor and X3D-Edit. Most tools are certified for the interchange profile and it is an easy task to retrieve this information and the detailed list of supported X3D components for each tool. Two of the authoring tools presented, Flux and Vivaty studio, are no longer active. Besides the information presented in the comparison by WEB3D consortium, a natural categorization for technologies regards their relation to the consortium that is and the details of the implementation. X3D-Edit relies on the NetBeans development platform and offers the Scene Access Interface (SAI) API in order to load and manipulate the scenegraph. BS Editor relies on a variety of technologies ranging from JAVA to proprietary ones and offers a software development kit (SDK) that can extend the capabilities of the tool. The core of the development kit also includes a scene manipulation API however, using the SDK is licensed. Blender is another interesting tool in the field of authoring released under the general public license. It offers a complete environment for model and scene manipulation, useful features for import/export, additional functionality such as game modes and full access to the source code.

C. Web advertising infrastructure

The component architecture of the iPromotion platform is presented in Fig. 1. The system implements a 3-tier serviceoriented platform composed of the following components:



```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE X3D PUBLIC 'ISO//Web3D//DTD X3D 3.2//EN' 'http://www.we
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.o
<head>
     <component level='2' name='CADGeometry'/>
</head>
<Scene>
<Viewpoint position=' 0 0 50' description='AuthoringTool'/>
<Viewpoint position=' 0 0 35' description='Initial' jump='true'/>
<Viewpoint position=' 35 0 0' orientation='0 1 0 1.57' descriptio
<Viewpoint position=' 0 35 0' orientation='1 0 0 -1.57' descripti
<Script DEF='MarineOScScript' mustEvaluate='true'>
                <field accessType='inputOnly' name='SensorInput'
                <field accessType='outputOnly' name='ScaleOutput'
                <![CDATA[ ecmascript:
                function SensorInput(value) {
                if (value[1]<0) {value[1]=-value[1];}
                ScaleOutput=new SFVec3f(value[1],value[1],value[1
]]>
</Script>
<Transform DEF='MarineOTr' translation='0 0 0'>
<Transform DEF='MarineORo' rotation='1 0 0 0'>
<Transform DEF='MarineOSpherePos' translation='0 0 0'>     <Group
         <Shape>
             <Box/>
             <Appearance>
                 <Material/>
             </Appearance>
         </Shape>
         <PlaneSensor DEF='MarineOSensorTr'/>
     </Group>
</Transform>
<Transform DEF='MarineOSphereRot' translation='3 0 0'>
     <Group>
        <Shape>
            <Sphere/>
            <Appearance>
                <Material/>
            </Appearance>
```

Fig. 2: the scene composition environment takes advantage of the X3D event model to implement scene management

- front-end services: based on HTTP, HTTP/SOAP and WebSockets, the system communicates with any type of device in order to adapt the rendering procedure to its capabilities using the mechanisms described in the previous paragraphs and also with any component within the architecture
- middleware container: it implements the fundamental framework that performs 3D rendering and interfaces with the presentation layer, the cloud that stores scenes, the MPEG-7 production service [8] and, last but not least, with high level tools such as 3D banners manager.
- cloud storage infrastructure: the repository of scenes and models repository is stored in a remote infrastructure implemented by as storage-as-a-service. The cloud supports transparent mechanisms for module search and retrieval over distributed processing technologies such as Hadoop and NoSQL databases [12].

Our work has been part of an online advertising platform infrastructure current under development under the iPromotion a research project. The aim of the project is to provide the all enabling framework and applications for online advertising based on VR technologies. The final outcome of the project is expected to support at least two high level use cases for online advertising. The first one will be targeting standard internet access from workstations, mobile or intelligent devices. Users will be provided with the opportunity to search online clouds of models and scenes and select any ones that interest them. Scenes will be transparently downloaded, presented transparently to the user who will be the capable to interact with the advertisement. The second use case will involve modern interaction interfaces that is, touch-screens, motion sensors such as the Microsoft KINEKT or ASUS XTION, and possibly wall screens or projectors. Users will be offered with VR environments conforming at least to the immersive X3D profile that supports 3D interactions and will thus be capable to interact with the scene using all the aforementioned interfaces. Both use cases will be supported by mechanisms for content-based search that take advantage of online MPEG-7 media location descriptors [4] for 3D scenes. MPEG-7 representation is also stored in the cloud along with the scenes. The transmission mechanism will conform with the MPEG-21 [5] standard in order to implement multimedia adaptation that is, to ensure that content can be played in the requesting devices. Finally, MPEG-4 is used to merge this information into unified 3D multimedia objects, incorporating information about digital and the actual scene description. Our infrastructures implement remote rendering of VR environments [10][11]. That is, 3D scenes are rendered on the server tier and end users receive video streams that have been adapted to the characteristics of their terminal devices as far as any characteristic is concerned (image quality, the frame rate). To this end, the open source Xj3D Browser has been ported to expose Axis2 Java Web service interface. A detailed description of the platform can be found in [13].

III. Composing Scenes

In order for the iPromotion platform to complete the cycle of 3D advertising production and service provisioning it is essential to support an authoring module that will allow scene authors or designers to produce models, scenes or worlds to be exposed by the platform. The authoring environment was required to be an autonomous desktop application environment but also to be able to exploit iPromotion services transparently that is, to use the scene repository cloud and the web services that facilitate scene creation, such as the services for scene description and retrieval. Considering parameters such as compliance to standards, technology excellence, development efficiency and organization we have chosen to use and extend the well-known X3D-Edit platform, already mentioned in section II that describes related work in the field. More specifically, X3D-Edit is released by the Web3D forum and is thus compliant with the spirit and the principle of the project. It is based entirely on JAVA and specifically on the NetBeans development platform, an important design characteristic since it offers all the benefits of the known window environment relieving the development process from the respective load while allowing efficient extension mechanisms through the creation of autonomous modules and plugins. X3D-Edit itself is modular by design since its codebase groups its functionality into distinct modules that group its functionality; KMLEditor, JOMWrapper, LookAndFeel, AppConfiguration, can be easily identified in terms of functionality and also in the code base, version 3.3, besides others. The tool is thus very easy to extend and also to compile since it exploits the build system of the NetBeans platform.

Fig. 2 presents the authoring tool extension of X3D-Edit that serves the necessary functionality for iPromotion. It is integrated within X3D-Edit as an autonomous plugin that exposes the functionality for iPromotion. The plugin implements a window viewpoint labeled “IPROMOTION” and is constructed as a separate NetBeans module, shown at the right in the figure. The panel on the left includes the port of the X3D viewer for JAVA. The main window of the tool is the X3D editor module used to compose X3D scenes and then load them in the viewer. The iPromotion panel presents a matrix containing all the scenes available from the cloud. The latter are selectable visual component so that they can be loaded. Users may also apply simple name matching in order to reduce the number of presented scenes using the input box bellow the scene matrix. Thus, after the wanted scene is located they are selected, fed in the viewer and then positioned as users will require. The new scene can be saved in the cloud transparently and can be loaded again for further processing. Each time a scene is saved, MPEG-7 indexing is transparently produced since the final scene output is submitted to the respective web service.

The main challenge behind the implementation of the aforementioned functionality is found in the inherent representation of scenes by X3D rendering modules, viewers. Such modules typically parse X3D input and do not discriminate among autonomous, integral 3D models that typically represent a single thematic object, which would be a very useful characteristic for our system as well as other ones. We have considered the alternatives for overcoming this problem; an elegant solution would be to construct a small framework that would be able to hold information regarding the state of the viewer that related to individual models being loaded even if there is no such notion in Xj3D rendering. The <x3d> tag is unique in every model and so is the <scene> tag. If the Xj3D viewer is to produce the outcome of model concatenation, they must first dispose the aforementioned tags, be processed and only then be fed to the viewer. This functionality can be implemented using both XML processing and the SAI API that supports methods for appending scenes in the one already presented by the viewer. In any case, the viewer would not be able to discriminate among the individual models it renders. Unless this functionality is included in the standard, the viewer can be set to expose information about each inserted model implicitly, event if it does not explicitly support such functionality. The solution we employed relied on visual controls in the form of basic shapes that have been inserted into the scene in order to provide feedback to the XJ3D runtime environment. Fig. 2 illustrates the visual dimension of the this mechanism; the cube, sphere and cone shapes behind the two scenes in the viewer, the car and the marine, are sensor nodes connected with the position, rotation and scale attributes of each scene with the aid of route elements in the first two cases and with the help of appropriate Javascript scripts in the latter. Thus, using standard X3D mechanisms, scenes are manageable and any environment can be built by adding more individual objects. This mechanism includes a number of details, the most important of which have to do with scene and sensor naming, listening to events, correct sensor positioning and then keeping the state of the viewer. Identifies must be named uniquely within scenes that is, all element definitions, DEFs, that relate to scene reference must be changed in real time. An efficient convention is to assign them their original names in addition to a suffix composed by the name of the scene and a number that counts how many instances of the scene have been loaded. This convention allows the authoring to maintain a structure, indexed by the unique scene instance names, holding the numbers that correspond to position, scale and rotation translations as returned by event listener mechanisms. Note however that, since the scene name is used as a key value to recover content it uses such as textures, images, etc, from the cloud, it must be maintained. Fig. 3 presents the implementation of the controls using X3D route elements that redirect numerical values from the sensors to the models they are to control using translation elements. As illustrated in the figure, the mechanism is implemented by means of X3D objects added in the scene. The names of Nodes correspond to the unique scene names suffixed with a string that identifies the sensor. Fig. 3 illustrates only the suffix for sensor node names however the implementation assigns names according to the aforementioned rule in order to implement control for any model. For the transformation of the size we employed the code illustrated in the script in order to convert vertical motion on the cone to scale information. Correct positioning of the sensors includes the computation of the maximum dimension of the scene in the Y-axis as measured from the center of the scene and the initialization of the position parameters for the sensors. This logic includes the identification of dimensions for large shapes and top level drawable objects and the transformations causing them to move, rotate, displace, etc.. The maximum dimension of the scene will define the position of the sensors so that they are visible and not hidden within space occupied by part of the object.

Fig. 4 presents parts of the output file that is stored in the cloud when a scene is produced. The format we have selected in order to incorporate composite multimedia objects based on 3D graphics is the MPEG-4 eXtensible MPEG-4 Textual format (XMT). XMT is XML-based described in MPEG-4 part 11 regarding scene description and application engine [14]. It allows us to describe the structure of multimedia scenes, by incorporating large parts of SMIL, SVG and X3D. Especially with respect to X3D, XMT uses an almost identical set of commands, making the two formats highly compatible. Besides representing simple or composite scenes, however, XMT also allows for the incorporation of more complex multimedia features, such as MPEG-7 descriptions and MPEG-21 rights management, all in the same structure. The top part of fig. 4 presents how scenes are described in terms of a world. The <es_descriptor> tag contains among other, the <streamsource> tag that defines the name of the scene which is also its index within the iPromotion cloud. Each <es_descriptor> defines a distinct scene that needs to be added in the viewer. The body of the XMT file groups all the scenes within the world within the <children> tag and keeps their position, rotation and scale as shown in the lower art of fig.4. This formalization allows scenes to be reused since complex scenes refer to simple, basic models and need not store additional X3D information. Models, that have not been created by combination of other 3D objects, do not need this kind of representation. For them the XMT information is absent from the cloud and it is never created; this is a means for separating composite and primitive scenes.

The mechanisms described in the previous paragraph are sufficient to allow the implementation of high level use cases that hide internal operations. Scenes are accessed through the iPromotion cloud API, and their manipulation relies on the SAI API. As mentioned in the previous paragraph, the implementation could rely on XML manipulation. However is an ISO standard and so is its binding to the JAVA language. We evaluated both approaches and empirically found that the former performed much slower than using the SAI API. Supplemental services, such as the creation and validation of indexing MPEG-7 information for the cloud, use relevant web services exposed by the middleware. Thus, users only deal with scene processing either graphical, or even textual through the X3D editor panel. Moreover, the tool exploits the editing characteristics X3D-Edit which supports all the X3D visual components. Another interesting characteristic is that the authoring tool is designed to allow the addition of textual meta-tags in the final scene. This information is incorporated into the MPEG-7 description and can thus be used for applying detailed search functionality. The searchable pieces of information will not be confined to textual meta-information; it will also include content descriptors for color, shape, texture and animation. As both the X3D and MPEG-7 standards are based on XML, the extraction of content description from scenes can be achieved through with XML-to-XML transformation.

IV. Conclusions And Future Work

The previous paragraphs describe a scene authoring and management tool designed to cover the needs of iPromotion, our virtual reality advertising platform. The primal use case that this software covers exceeds the basic scene creation and editing; the functionality we aim to deliver aims to provide o higher layer in 3D authoring by focusing on the concatenation of completed, autonomous models and/or scenes in order to create manageable 3D worlds. We have extended a valuable scene composition tool in the field, X3D-Edit, that relies on the Netbeans platform to offer scene manipulation functionality in moduls. We thus added a new window in the top level view of the tool that integrates iPromotion functionality transparently, more specifically, we employed the programming interfaces for cloud access and scene indexing. The former required the support of proper representations for the new types of scenes and the latter required the construction and communication with appropriate web services that extract indexing information. The core of the system however is the scene manipulation logic that keeps the state of the worlds constructed by users, that is the identification of individual scenes their normalization so that they can be fed into the XJ3D browser. Thus, a scene can be added multiple times; information about each instance is kept and all of them will be capable to access any resources originally available to them such as textures, images, audio, etc.. Scenes are controlled by visual objects properly positioned below them. These controls are accompanied with X3D and Javascript code that transforms their events to scene positioning, rotation and scale. The described work is currently in alpha stage that is, it is in progress. Future work will focus on all aspects of scene authoring. Since XJ3D browsers currently support access to online resources, scene elements using the cloud interface can be limited only to the necessary extend. The tool is following the evolution of the iPromotion VR platform and is expected to be tested against more elaborate models representing 3D environments. It will be evaluated for performance and usability.

Acknowledgments

The work described in this publication has been funded by the Programme “Cooperation” of the Operational Programme “Competitiveness and Enterpreneuship” (OPCE II) of the

Ministry of Development and Competitiveness, partly funded by the European Regional Development Fund, 2007-2013.

REFERENCES

[1] David Mazursky and Gideon Vinitzky, Modifying consumer search processes, ISSN 1860-2037 in enhanced on-line interfaces, Journal of Business Research 58 (2005), no. 10, 1299 – 1309.

[2] Kil-Soo Suh and Young Eun Lee, The effects of virtual reality on consumer learning: an empirical investigation, MIS Quarterly 29 (2005), no. 4, 673–697.

[3] Serge Baile, Sana Debbabi, and Mohamed Daassi, Effect of online 3D advertising on consumer responses: the mediating role of telepresence, Journal of Marketing Management (2010), no. 2, 1472–1476.

[4] Bangalore S. Manjunath, Phillipe Salembier, and Thomas Sikora, Introduction to MPEG-7: multimedia content description interface, John Wiley & Sons, Inc., New York, NY, USA, 2002.

[5] Ian S. Burnett, Fernando Pereira, Rik van de Walle, and Rob Koenen, The MPEG-21 book, John Wiley & Sons, the remote renderer: for higher quality scenes, to be 2006.

[6] WORLD WIDE WEB CONSORTIUM – W3C (2001), Scalable Vector Graphics (SVG) 1.0 Specification, http://www.w3.org/TR/SVG10/, (Accessed 19 May 2010)

[7] WEB3D CONSORTIUM (2004), Extensible 3D (X3D) ISO/IEC http://www.web3d.org/x3d/specifications/#x3d-spec, (Accessed 19 May 2010)

[8] P. Spala, A. G. Malamos, A. D. Doulamis, and G. Mamakis, "Extending MPEG-7 for efficient annotation of complex web 3D scenes," Multimedia Tools Appl, vol. 59, no. 2, pp. 463–504, 2012.

[9] M. Zampoglou, P. Spala, K. Kontakis, A. G. Malamos, and J. A. Ware, "Direct mapping of x3d scenes to mpeg-7 descriptions," in Proceedings of the 18th International Conference on 3D Web Technology June 20-22, 2013 San Sebastian, Spain, accepted for publication.

[10] F. Lamberti and A. Sanna, "A streaming-based solution for remote visualization of 3D graphics on mobile devices," IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 2, pp. 247–260, 2007.

[11] T. K. Capin, K. Pulli, and T. Akenine-M¨ller, "The state of the art in mobile graphics research," IEEE¸ Computer Graphics and Applications, vol. 28, no. 4, pp. 74–84, 2008.

[12] G. Kousiouris, G. Vafiadis, and T. A. Varvarigou, "A front-end, hadoop-based data management service for efficient federated clouds," in Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, pp. 511–516, 2011.

[13] Markos Zampoglou, Athanasios G. Malamos, Kostas Kapetanakis, Konstantinos Kontakis, Emmanuel Sardis, George Vafiadis, Vrettos Moulos and Anastasios Doulamis. "iPromotion: A Cloud-Based Platform for Virtual Reality Internet Advertising", Big Data and Internet of Things: A Roadmap for Smart Environments, Studies in Computational Intelligence 546, DOI: 10.1007/978-3-319-05029-4_19, Springer International Publishing Switzerland 2014

[14] Fernando Pereira and Touradj Ebrahimi. The MPEG-4 book. Prentice Hall Professional, 2002.

Michael Kalochristianakis is currently an associate lecturer and researcher at the Technological Educational Institution of Crete (TEIC), at the Department of Informatics Engineering. In the past I have been an associate lecturer for the Dpt of Applied Science at TEIC, for the Dpt of Production and Engineering Management at the Technical University of Crete and for the Academic Library at the University of Crete. He has also worked for the Research Academic Computer Technology Institute and Press "Diophantus" as an IT researcher/engineer and for the software industry as a J2EE programmer. Michael holds a Doctorate degree in Computer Engineering and Informatics, a Masters degree in Computer Science and a Diploma in Electrical Engineering and Computer Technology. Pieces of my work have been published in international peer-reviewed journals and have been presented at international conferences. He is also a reviewer for several academic journals.

Markos Zampoglou was born in 1981 in Thessaloniki, Greece. He graduated from the dept. of Applied Informatics of the University of Macedonia, Greece, in 2004. He received an MSc in Artificial Intelligence from the University of Edinburgh in 2005 with a Distinction. He was awarded a Doctorate degree from the dept. of Applied Informatics, University of Macedonia, Greece in 2011. Markos is currently an associate lecturer and researcher at the Technological Educational Institution of Crete (TEIC), at the Department of Informatics Engineering.

Kostas Kontakis is a graduate student at the Technological Educational Institution of Crete (TEIC), Department of Informatics Engineering pursuing his MSC. His primary interests include semantic web and ontologies. He has been with the Multimedia Content Lab for two years. Kostas holds a BS degree from the Department of Informatics Engineering.

Kostas Kapetanakis is an experienced systems administrator and programmer working for the Multimedia Lab. He is also graduate student at the Technological Educational Institution of Crete (TEIC), Department of Informatics Engineering. He is currently pursuing his MSC degree. Kostas holds a BS degree from the Department of Informatics Engineering.

Athanasios G. Malamos received a BSC degree in Physics from the University of Crete (1992) and a PhD from the Technical University of Crete in 2000. From 1997 to 2002 he was a research assistant and a researcher in the ICCS National Technical University of Athens. Since 2002 is with the Technological Educational Institute of Crete, at the Department of Informatics Engineering, as an Assistant Professor (2002-2006) and as an Associate Professor (2006 until present). Prof. Malamos leads the Multimedia Lab. He has been running EU and National funded research projects. He has served as program committee and reviewer for several international conferences and workshops. He is a reviewer for IEEE, Springer as well as for international journals that fall within his interests and he is a member of the WEB3D consortium. His research interests include multimedia services, virtual reality, 3D modeling and multimedia semantics.