

# Virtual Training for Manufacturing and Maintenance based on Web3D Technologies

Demis Corvaglia

HCI Lab, Dept. of Math and Computer Science

University of Udine

via delle Scienze 206, 33100 Udine, Italy

+39 0432 558413

[corvagli@dimi.uniud.it](mailto:corvagli@dimi.uniud.it)

## ABSTRACT

In this paper, we (i) introduce the Computer-Based Training (CBT) context, in particular CBT that employs Virtual Reality (VR) technologies; (ii) describe the generic environment we developed to allow the employment of immersive hardware in Web3D applications, (iii) describe the immersive Virtual Training (iVT) application we have built for the training of industrial maintenance personnel.

## Categories and Subject Descriptors

H.5.2 [User Interfaces]: *Training, help, and documentation*. I.3.6

[Methodology and Techniques]: *interaction techniques*. I.3.7

[Three-Dimensional Graphics and Realism]: *virtual reality*.

## General Terms

Algorithms, Management, Documentation, Design.

## Keywords

Computer-Based Training, Virtual Manufacturing, Virtual Manuals, VRML, X3D, Virtual Reality Environment.

## 1. INTRODUCTION

Well-designed Computer Based Instruction (CBI) and Training (CBT) application have demonstrated to be effective as learning tools (see, e.g., [15][18][10]). Virtual Reality (VR) is used to improve realism and effectiveness of CBT systems in major fields such as medicine and disabled-care, military and industrial applications. In medicine, VR allows to train surgeons to perform medical operations without risks for patients [16][5]. Moreover, VR training helps the disabled to learn how to deal with every day tasks, avoiding dangerous situations or social discomforts [8]. In military applications, soldiers can virtually try extreme scenarios [24][3] and practice with handling operations in foreign cultures [9]. Since 1991, NASA employs VR [28] where space missions can be tested and astronauts can learn shuttle functionality. Industry, e.g. automotive, is also strongly interested in VR for the training of design, production and maintenance [2] [24]. Sense of Presence (SoP) [20] and interaction richness [17] are primary requisites of a VR training system. To improve these aspects, virtual characters are an effective tool in real world applications where users have to learn hand-operated tasks. Virtual training systems such as Jack [6] and Steve [27] adopted this approach.

In general, VR training offers powerful tools, but it has to be both graphically and physically realistic and comfortable (ergonomic) for trainees.

This paper presents a CBT system for the manufacturing and maintenance contexts based on VR hardware and Web3D technologies. In the considered context, trainees need to interact with mechanical parts, using specific tools (e.g. snapper, screwdriver, ...), that can be virtually simulated. VR (especially immersive VR) is here able to: (i) provide an appealing training environment, (ii) avoid dangerous situations, and (iii) reduce training costs [19].

The paper is organized as follows. Section 2 introduces iVRML (immersive VRML), a generic environment we developed to allow the employment of immersive hardware in Web3D applications. We use iVRML to develop the iVT (immersive Virtual Training) system described in Section 3. Section 4 presents an example of iVT applied to jet engine maintenance. Section 5 summarizes the main results of this project, limitations and future work.

## 2. iVRML (immersive VRML)

VRML (Virtual Reality Modeling Language) [29] is a language for the definition of interactive 3D graphics contents both for standalone applications and for the Web. VRML became an ISO standard in 1997, and the X3D (the language that evolved from VRML) specification [30] has recently been approved by ISO.

iVRML is a generic framework to provide an immersive experience using standard VRML/X3D worlds. iVRML allows to:

- use VR hardware (e.g. HMDs, datagloves, 3DOF/6DOF sensors) in VRML/X3D worlds;
- easily develop specific VR applications, such as the iVT (immersive Virtual Training) described in detail in Section 3.

Section 2.1 briefly introduces the iVRML software architecture and user interface. Section 2.2 discusses VR hardware compatibility and the interface between device drivers and VRML world. Section 2.3 presents an example of a basic iVRML framework extension: user navigation in 3D worlds using hand gestures and head movement.

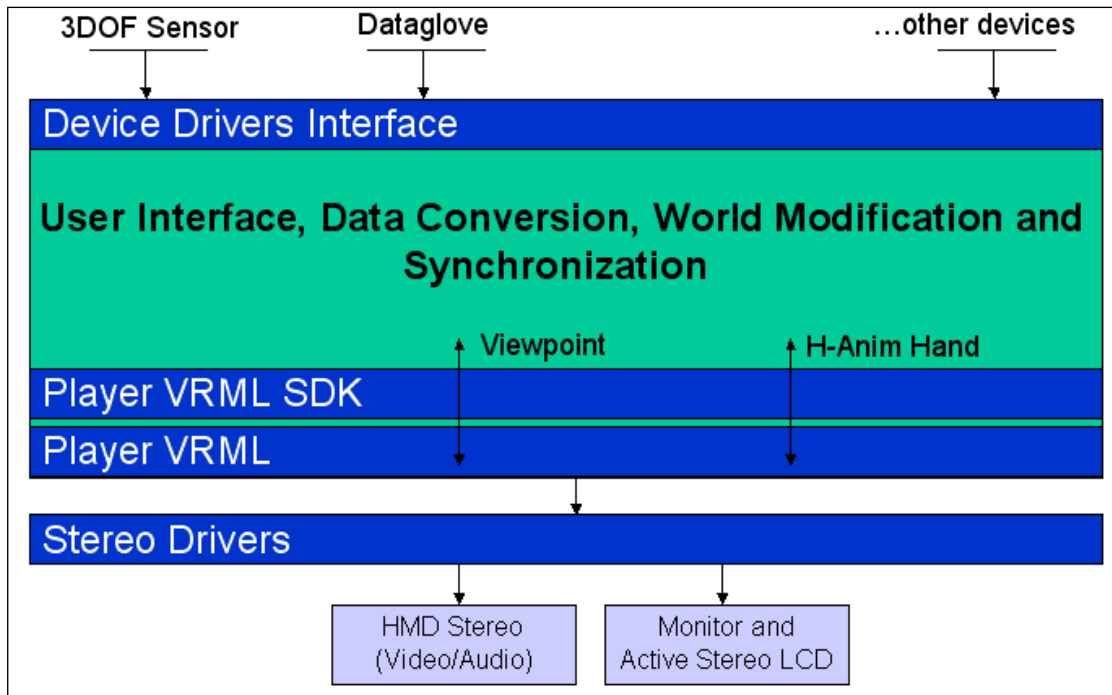


Figure 1. High level iVRML architecture.

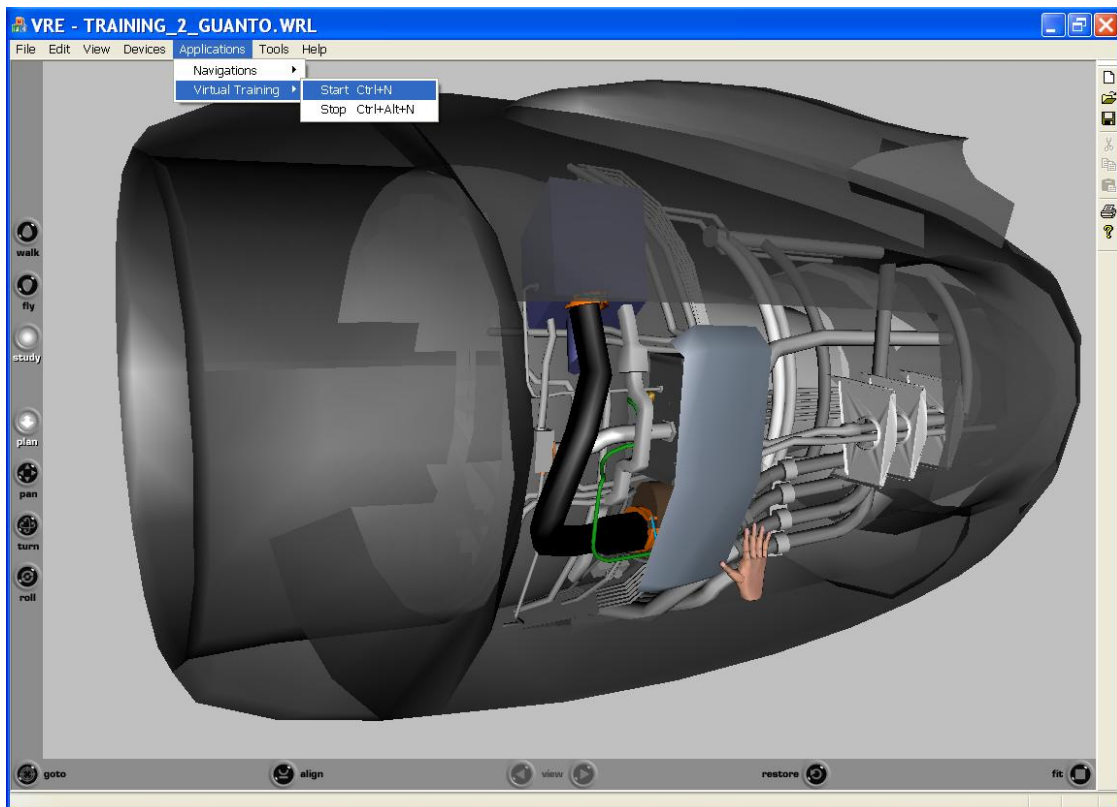


Figure 2. iVRML user interface during a training session on a jet engine (engine model downloaded from [23]).

## 2.1 ARCHITECTURE AND USER INTERFACE

iVRML architecture is shown in Figure 1. The main container (the biggest box) is a Visual C++ application. It receives data from VR hardware (e.g. 3DOF/6DOF sensors, datagloves); this data is processed by the Device Driver Interface (DDI) that provides methods to access data structures updated by VR devices. First, the application core converts and integrates data from DDI (automatically or under user directives inserted by UI); second it updates the VRML world depending on the application (simulations are computed in reaction to user stimuli). The VRML world is displayed in the main frame of the application, through a Cortona [22] VRML player ActiveX control. VRML scene tree manipulation is performed using Cortona SDK. Common iVRML functions are:

- changing viewpoint orientation according to 3DOF sensor output;
- driving virtual hand according to 6DOF sensor and dataglove (fingers flexure) output;
- detecting and responding to collisions between user controlled objects and other scene objects.

These functions are configurable depending on specific application goals.

The last layers of the diagram shown in Figure 1 concern Stereo Drivers and displays. They allow stereoscopic graphics output. To implement this architecture layer we employed nVidia Stereo Drivers [21] coupled with a GForce graphics board. Stereo video signal is driven to Head Mounted Displays (HMD) for immersive experiences or to CRT monitors (in this case, stereo effect is obtained using active LCD stereo glasses).

Figure 2 shows an iVRML user interface. The application menu allows users to customize basic application behaviour. It concerns:

- opening VRML files;
- activating and setting VR devices;
- connecting VR device data to VRML entities (e. g. 3DOF sensor to a viewpoint);
- enabling special iVRML functionality support, such as object-to-object collision detection, text-to-speech, positional audio and environment sound effects (e.g. reverb).

When user chooses a specific iVRML application (for example, Figure 2 shows “Virtual Training” activation), stored configurations are loaded and iVRML displays dialog windows specifically designed for current task. Applications can extend the specific user interface even in the VRML world using 3D widgets (e.g., 3D buttons or slides, On Screen Displays, ...), indeed, VRML events can be connected to application core management.

## 2.2 VR HARDWARE

So far, we have used iVRML connected to several VR hardware devices, in particular:

- 3DOF sensors (interTrax2 and inertiaCube2 from interSense [13]);
- datagloves (Dataglove 5 from 5DT [1]);

- Head Mounted Displays (HMD) based on the Cy-Visor platform [7].
- Active LCD stereo glasses (Crystal Eyes 3 from StereoGraphics [25]).

iVRML allows one to extend device support easily thanks to DDI (see section 2.1), e.g. to employ 6DOF sensors such as the pciBird from Ascension [4]).

## 2.3 iVRML NAVIGATION MODES

In 3D virtual worlds, navigation is often the main task. VRML\X3D players allow different navigation modes [30], but all of them concern only user interaction through keyboard, mouse or joystick. Using VR hardware, navigation modes can be improved and tailored to a more immersive experience. In iVRML, we have implemented two major navigation modes:

- **Guided:** user’s viewpoint is tied to an animated scene object (e.g. virtual car); viewpoint orientation is also influenced by 3DOF sensor on HMD.
- **Free:** body position/orientation and velocity are controlled by hand gestures, that are captured by a dataglove (or a joystick); viewpoint orientation is controlled both by body position/orientation and head orientation (this way, the user can move towards a given direction while looking towards a different one). This mode is implemented by combining body position/orientation and head orientation.

Figure 3 and Figure 4 show the two navigation modes. In Figure 3, the user is in Guided mode on board of an aircraft and he is experiencing the manouvres of an aerobatic team [11]. In Figure 4, the user is in Free navigation mode, exploring a cultural heritage reconstruction [26].



**Figure 3. Guided navigation mode. User position is constrained (e.g. inside an aircraft); Viewpoint orientation changes following user’s head (3DOF sensor mounted on HMD).**



**Figure 4. Free navigation mode. Body position, velocity and orientation are driven by hand gestures; viewpoint follows user head orientation (as in the real world).**

### 3. The iVT (immersive VIRTUAL TRAINING) application

iVT (immersive Virtual Training) is a CBT system implemented over iVRML. Industrial maintenance procedures are often very well defined and tasks sequence can be very elaborate. The learning process is effective if users can directly work on physical components. iVT provides this opportunity without risks (both for the maintainer and for the components). iVT assists the trainee during training sessions; it suggests the correct task to perform through a textual description (optionally converted into audio by a text-to-speech engine). Current iVT implementation supports right hand interaction based on a dataglove and a tracking sensor mounted on it. When iVT suggests the task to perform, the user can move her virtual hand and execute the task. Collision detection between virtual hand and scene objects is computed in real-time. The hand structure follows the H-Anim specification [12]; it is provided with spherical collision detection sensors (see Figure 5) that are used to determine hand-object proximity or contact.



**Figure 5. Spherical sensors of the virtual hand.**

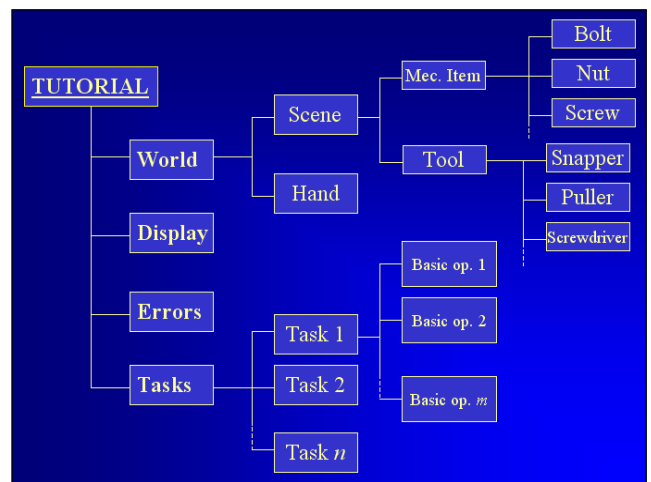
iVT offers low level support to develop virtual training application. iVT administrator (i.e. the person who defines the tutorial) instantiates and extends a language whose syntax is standard VRML and whose structures is shown in Figure 6.

We have designed specific PROTOs to describe:

- **Mechanical Items:** bolts, nuts, screws, ...;
- **Tools:** snappers, pullers, screwdrivers, ...;
- **Training Tasks:** a sequence of basic operations;
- **Basic Operations:** user's actions and associated conditions (temporal and spatial) checked by iVT.

An example of a task with some basic operations is the following: the trainee has to remove a metal cover from a jet engine (see Figure 7), she has to move her virtual hand towards the item (basic operation 1) and grab it (basic operation 2). iVT tests the correct grab conditions and when these are satisfied, a visual effect advises user. After this, hand and mechanical item have to move together, both controlled by the dataglove. When the user reaches the required release position (basic operation 3), the item disappears and the system indicates the new task to perform.

Every mechanical item has its specific assembling/disassembling conditions and associated mechanical tool. iVT is developed for common mechanical tasks, but the administrator can define new ones. For example, if a mechanical item needs a special tool, the administrator has to define item and tool geometry, start position of the item inside the complete mechanic part, insertion and extraction conditions for the item, tool and item interaction rules. This definition can involve complex interaction that requires to write ad-hoc javascript code. However, for learning purposes, it is sufficient to define item-tool(s) association, point of interaction and the tool-hand(s) grabbing conditions. When a correct tool-item (relative) position is reached, tool usage can be demonstrated by 3D animation.



**Figure 6. Language abstract diagram designed to define a tutorial in iVT.**

A (human) trainer (that can be the administrator or a high level supervisor) can define messages that will be displayed during the training, and for error conditions and warnings.

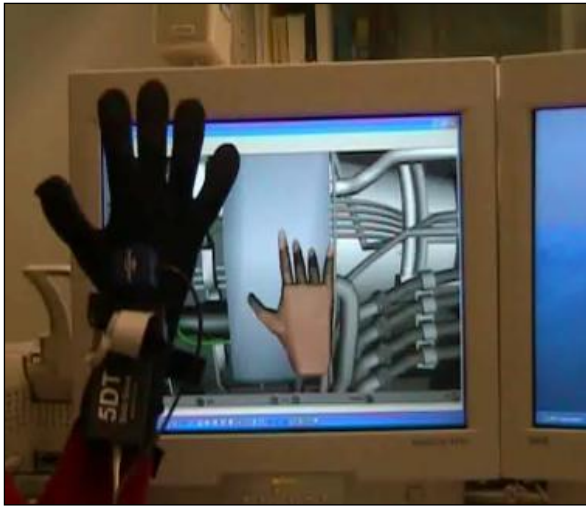


Figure 7. A iVT user is learning how to remove a cover from the jet engine.

#### 4. iVT CASE STUDY: DISASSEMBLING AN AIRCRAFT JET ENGINE

We have experimented iVT on the 3D model and disassembly procedures available at [23]. While the desktop version of the virtual manual is limited to showing the tasks to the user, our re-implementation in iVT allows the user to virtually try the disassembling phase with her hand. Figure 8 and Figure 9 show two training examples. The first requires to unscrew a bolt to free a cover; iVT uses an On Screen Display (OSD) to tell the user how to perform this task. This task is followed by cover removing shown in Figure 7. In Figure 9, the user removes a tube; this task requires correct hand posture.



Figure 8. Unscrew the bolt (using “finger-screwdriver”).

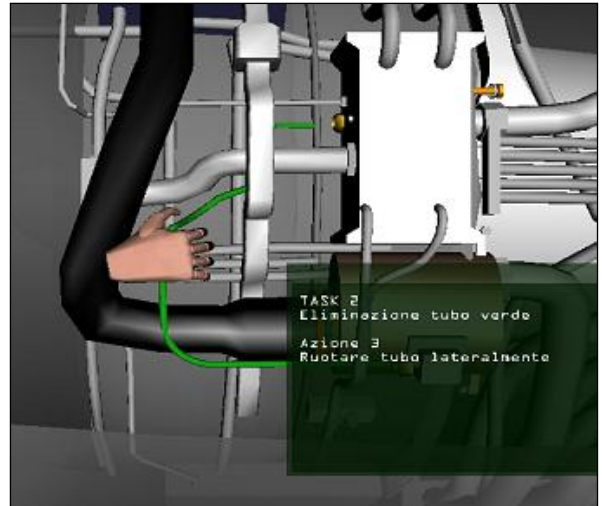


Figure 9. Green tube removal. The OSD tells to user: “Turn tube sideways”.

#### 5. CONCLUSIONS

In this paper, we presented mainly our iVRML and iVT applications. Informal user testing sessions produced encouraging results about iVT potential. Users greatly appreciated the possibility of virtually trying in a direct way the actions that are usually only displayed in virtual manuals. We plan to perform controlled experiments when more iVT functionalities will be added. Indeed, there is a number of limitations of the system on which we are working:

- enabling two-handed interaction;
- enabling force-feedback and haptic devices support;
- implementing object-to-object collision detection;
- designing and implementing a visual user interface for authoring training materials.

An important future development will consist in the introduction of virtual characters that demonstrate correct tasks execution to users.

#### 6. ACKNOWLEDGMENTS

Elisa Cazzitti played an important role in the implementation and testing of the first version of the jet engine application.

#### 7. REFERENCES

- [1] SDT home page: [www.5DT.com](http://www.5DT.com).
- [2] A. Gomes de Sà, G. Zachmann, *Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes*. *Computer & Graphics*, Elsevier, Vol. 23, No. 3, pp. 389-403, 1999.
- [3] A. W. Stedmon, *Re-viewing Reality: Human Factors of Synthetic Training Environments*. *International Journal of Human-Computer Studies*, No. 55, pp. 675-698, 2001.
- [4] Ascension Technology home page: <http://www.ascension-tech.com>.
- [5] B. Fröhlich, G. Grunst, W. Krüger and G. Wesche, *The Responsive Workbench: a Virtual Environment for*



- Physicians*. Comput. Biol. Med., Vol. 25, No. 2, pp. 301-308, 1995.
- [6] Badler N. I., Philips C. B., Webber B. L., *Simulating Humans, Computer Graphics Animation and Control*, Oxford University Press, New York 1993.
- [7] Cy-Visor HMD: <http://www.personaldisplay.com>.
- [8] D. Gourlay, K.C. Lun, Y. N. Lee and J. Tay, *Virtual Reality for Relearning Daily Living Skills*. International Journal of Medical Informatics, No. 60, pp. 255-261, 2000.
- [9] E. Sims, *Using Extensible 3D (X3D) to Repurpose Modeling and Simulation Assets for Distributed Learning*. Technical Report, [www2.vcom3d.com/SISOSpring03.pdf](http://www2.vcom3d.com/SISOSpring03.pdf).
- [10] F. Mantovani, *VR learning: Potential and Challenges for the Use of 3D Environments in Education and Training*, Chapter 12 in *Towards Cyber-Psychology: Mind, Cognitions and Society in the Internet Age*, Amsterdam, IOS Press, 2001.
- [11] Frece3D web site: [www.frece3D.uniud.it](http://www.frece3D.uniud.it).
- [12] H-Anim (Humanoid Animation Working Group) home page: [www.hanim.org](http://www.hanim.org).
- [13] InterSense home page: [www.intersense.com](http://www.intersense.com).
- [14] J. R. Wilson, *Virtual Environments Applications and Applied Ergonomics*. Applied Ergonomics, No. 30, pp. 3-9, 1999.
- [15] J.D. Fletcher, *Does the Stuff Work? Some Findings from Applications of Technology to Educational and Training*. Proc. of Conference on Teacher Education and the Use of Technology Based Learning Systems, 1996.
- [16] K. Brodli, N. El-Khalili and Y. Li, *Using Web-based Computer Graphics to Teach Surgery*. Computer & Graphics, No. 24, pp. 157-161, 2000.
- [17] L.Sastry, D.R.S. Boyd, *Virtual Environments for engineering applications*, Virtual Reality 3 (1998) p.235-244.
- [18] M. Barnes, *Virtual Reality and Simulation*. Proc. of the 28<sup>th</sup> Conference on Winter Simulation, Colorado, California, pp.101-110, 1996.
- [19] M. Ponder, B. Herbelin, T. Molet, S. Schertenlieb, B. Ulicny, G. Papagiannakis, N. Magnenat-Thalmann, D. Thalmann, *Immersive VR decision training: telling interactive stories featuring advanced virtual human simulation technologies*. 9th Eurographics Workshop on Virtual Environments IPT/EGVE2003, pp. 97-106, Zurich, Switzerland, 2003.
- [20] M.Slater, *Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire*, Presence 8 (1999) p.73-79.
- [21] NVidia home page: [www.nVidia.com](http://www.nVidia.com).
- [22] Parallel Graphics home page: <http://www.parallelgraphics.com>.
- [23] Parallel Graphics, Virtual Manuals Gallery, <http://www.parallelgraphics.com/virtual-manuals/gallery/engine2>.
- [24] R. Stone, *Virtual Reality for Interactive Training: an Industrial Practitioner's Viewpoint*. International Journal of Human-Computer Studies, No. 55, pp. 699-711, 2001.
- [25] StereoGraphics home page: [www.stereographics.com](http://www.stereographics.com).
- [26] Udine3D web site: [www.udine3D.uniud.it](http://www.udine3D.uniud.it).
- [27] W. L. Johnson, and J. Rickel, "Steve: An Animated Pedagogical Agent for Procedural Training in Virtual Environments", Sigart Bulletin, ACM Press, vol. 8, number 1-4, 16-21, 1997.
- [28] W. R. Sherman, *Experiences with Virtual Reality Applications*. Proc. 24<sup>th</sup> Annual Conference on Computer Graphics and Interactive Techniques, pp. 473-476, 1997.
- [29] Web3D Consortium home page: [www.web3D.org](http://www.web3D.org).
- [30] X3D specifications: <http://www.web3d.org/x3d/specifications/index.html>.