BlenderVR: Open-source Framework for Interactive and Immersive VR

Brian F.G. Katz* Dalai Q. Felinto Damien Touraine David Poirier-Quinot Patrick Bourdot

LIMSI-CNRS, Campus Universitaire d'Orsay, Orsay, France

Abstract

BlenderVR is an open-source project framework for interactive and immersive applications based on an extension of the Blender Game Engine to Virtual Reality applications. BlenderVR is a generalization of the BlenderCAVE project, accounting for alternate platforms (e.g., HMD, video-walls). The goal is to provide a flexible and easy to use framework for the creation of VR applications for various platforms, making use of the existing power of the BGE's graphics rendering and physics engine. Compatible with 3 major Operating Systems, BlenderVR has been developed by VR researchers with support from the Blender Community. BlenderVR currently handles multi-screen/multi-user tracked stereoscopic rendering through efficient low-level master/slave synchronization process with multimodal interactions via OSC and VRPN protocols.

Index Terms: H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities; I.3.2 [Graphics Systems]: Distributed/network graphics;

1 Introduction

We present here the current state of development of the BlenderVR project.[1] BlenderVR is a generalization of the previous BlenderCAVE and BlenderCAVE3D-s projects [7, 6]. BlenderVR is principally a scene graph editor based on the well established Open-source Blender[2] and Blender Game Engine (BGE) [3] software. Blender is a multi-platform open source 3D creation content software with enough functionalities to create photorealistic pictures, high quality animations and, most of all, video games. Blender based games exploit the BGE real-time rendering engine which handles a multitude of physical interactions through the implemented Bullet Physics Library while general game logic may be defined through blocks and/or embedded python scripts. Gathering users for more than a decade, Blender now boasts a large support community, a dedicated professional networkand several scene repositories where one may find plenty of reusable material.

2 Main Features

BlenderVR, a patched version of the core Blender software, adds additional functionalities key to VR applications, while benefiting from the basic BGE architecture, interface, and user community. The BlenderVR code project is accessible via GitHub[3] and collaborators are welcome. The project website hosts documentation, executable downloads, and examples.

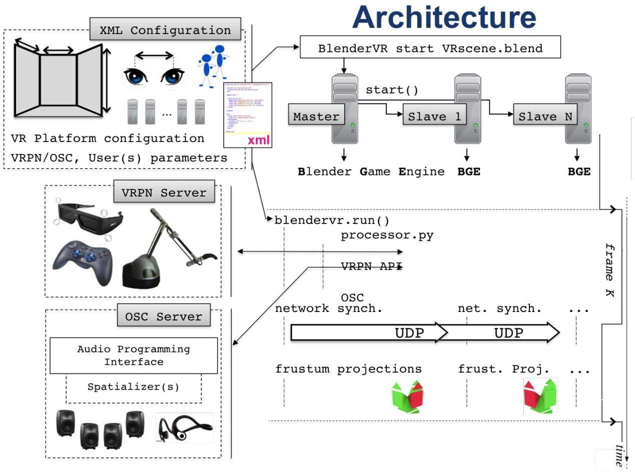

Figure 1: BlenderVR architecture graph showing (right) the 3 main processes integrated in the BGE frame based sequencer. *network synchronization* process maintains consistency between master/slave rendering nodes. *frustum projection* handles tracked adaptive stereoscopic rendering calculations.

The core BlenderVR modifications to the BGE master consist in the addition of a `prerender` method prior to the basic `predraw`, the ability to redefine the projection matrix, and the possibility to redefine the aspect-ratio from a python script (allowing easier portability of scenes between architectures). BlenderVR routines, as well as most of the BlenderVR patched modifications, have been implemented so as to be as transparent as possible regarding BGE native processes in the hope that they will soon be integrated in the official Blender master, eliminating the need for patching.

2.1 Master/Slave synchronization

The synchronisation of multiple graphic nodes is necessary for assuring continuity and coherence of the scene in situations with multiple displays (e.g., CAVE, videowall) or rendering instances (e.g., mutliple HMD). Master/Slave synchronization is carried out at each frame, in the `prerender` method. Synchronization, executed via a python script, inspects every object in the scene to see if it has changed. If so, the update information is passed from the Master to the Slave nodes before rendering. TCP multi-unicast is used for synchronization and acknowledgment messages. By default, BlenderVR synchronizes every object in the scene amongst the rendering nodes. BlenderVR also supports synchronization, white and black lists, forced synchronization without checks or no synchronization or checks for specific objects, thereby reducing unnecessary calculation overhead in complex scene rendering.

Already evaluated for CAVE-type configurations, current development concerns collaborative work with combined HMD and CAVE or multiple HMD devices on shared or remote sites. The importance of synchronization of scene graphs is crucial for such applications and BlenderVR should be well suited to this situation.

2.2 Adaptive stereoscopy for large-screen projections

Adaptive stereoscopy is required for Workbench, projected Walls or CAVE-like systems. This rendering mainly consists in changes of coordinate systems and projection operations. The resulting modification in the projection matrix is necessary because the user is, more often than not, away from the center of the screen and

*e-mail: brian.katz@limsi.fr

IEEE Virtual Reality Conference 2015
23 - 27 March, Arles, France
978-1-4799-1727-3/15/$31.00 ©2015 IEEE

thereby requires non-symmetric projection matrices. This is updated to take into account user's current head position, orientation, and eye separation. Such variable frustum projections have been implemented so as to be computed locally on each graphical node before rendering in the `prerender` process. This implementation is generic enough to manage multi-stereoscopy, which allows for several tracked users to each have correct depth perception.

2.3 External message processing

While stereoscopic rendering and synchronization are supposedly basic VR features, message exchange with external software represents a cornerstone in any scene graph editor as it largely impacts end users in their scene developments. BlenderVR to external (and vice versa) user defined interactions are conventionally implemented and collected in a python script attached to the VR scene. Incoming messages are often dedicated to user integration or interaction in the VR scene, from position/orientation coordinates to various controller states, and typically employ the VRPN protocol. For this, BlenderVR implements the *Processor* python Class which integrates dedicated OSC and VRPN APIs to ease message processing for access to controllers and user parameters.

Most outgoing messages are currently intended to add sound to the scene, i.e., update scene graph object properties in the Sound Rendering Engine (SRE) using the well known Open Sound Control (OSC) protocol. The SRE here refers to the system gathering audio object definitions (sound source, position, etc.) and sound rendering methods in an Audio Programming Environment (e.g., Max/MSP, PureData, etc.) plus eventual loudspeaker/headset outputs. The OSC API encapsulates three main Classes of messages: Global, Object, and User. A fourth class of message manages the routing of Objects to multiple spatialization engines (i.e., Users), such that certain Objects can be rendered for selected Users if desired, and is thus named the *ObjectUser* Class. These messages define which engine handles a given set of objects in the scene graph.

3 Example Installations

Scene graph edition in BlenderVR is essentially based on Blender creation and animation tools, while audio rendering related events are written in the Processor script. The overall process being architecture independent, most scene development can be carried out on standard laptops, before being ported to the actual VR systems. Once installed, basic import of a Blender scene (without tracking nor audio rendering) is almost transparent.

BlenderVR has been developed on two VR platforms: SMART-I2 [8] and EVE[4]. Portability of developed scenes between such different VR architectures is the driving goal behind the project.

The SMART-I2 implements passive adaptive stereoscopic rendering through a pair of front-projected rigid screens which also serve as a 24 channel loudspeaker array offering horizontal Wave Field Synthesis audio rendering. One computer handles graphics, with 4 projectors for 2 screens, running Ubuntu Precise on an Intel Core 2 2.66 GHz Quad Q9400, 4 GB of RAM, and 2 NVIDIA GeForce GTX 470. Audio is handled separately.

The EVE system offers a multi-user/multi-sensorimotor environment with adaptative double-stereoscopic rendering, 15 loudspeakers and 2 RF modules for wireless audio input and individual binaural renderings. EVE comprises 4 rear-projected screens coupled to 7 projectors providing $\sim 60\text{m}^2$ of high definition projection space. The system comprises 8 i7 computers (1/projector + 1 monitoring console) with Ubuntu Precise, 12 GB of RAM, and Quadro 6000. Audio and haptics are handled separately.

4 Example applications

In a study investigating paradigms for 6DoF navigation in immersive virtual worlds, [2] compared joystick-based input devices and steering metaphors based on movements of the user's body, e.g., head-controlled paradigms. BlenderVR served as the software platform, with various input devices controlling the user's flight through a series of navigational tasks. While the virtual world was static, the selection of various test configurations and generation of experimental logs were all achieved within BlenderVR.

To evaluate designs for a user audio guidance system for rescue workers, [5] implemented virtual prototypes to provide ecologically valid test conditions. The BlenderVR based virtual prototypes exploited various input devices (e.g., 6 DoF tracking, Wii Balance Board, Wiimote) along with OSC communication with MaxMSP to generate interactive audio feedback sonification based on geometrical data from the BlenderVR scene.

Real-time animation of virtual avatars based on motion capture within the BlenderVR framework was investigated by [4]. A Python module was developed to control avatars using an ART MoCap tracking system, common in VR applications.

Focusing on collaborative situations, [1] investigated navigation in a multi-stereoscopic immersive system (several users sharing the same restricted workspace). In this context, a proper navigation paradigm should provide users both efficient control of virtual navigation and a guaranty of users' safety in the real workspace relative to the display system and between users.

5 Conclusion

BlenderVR project was created in response to the lack of affordable scene creation software dedicated to Virtual Reality in augmented environments by implementing a new scene graph editor based on the popular Blender Game Engine. To adapt the BGE to VR, BlenderVR includes multi-user tracked stereoscopic rendering, master-slave synchronization, and OSC and VRPN methods for external applications. Windows, Linux, and MacOS compatible, BlenderVR is released as open source, for those in need of straightforward scene development solutions for multimodal VR creation.

Acknowledgements

This work was funded in part by the ANR Equipex DIGISCOPE project, the FUI-BiLi project (http://www.bili-project.org/), and an ANRT CIFRE industry-linked PhD research with Airbus D&S.

References

[1] W. Chen, N. Ladeveze, C. Clavel, D. Mestre, and P. Bourdot. User cohabitation in multi-stereoscopic immersive virtual environment for individual navigation tasks. In *IEEE Virtual Reality (IEEE VR)*, page (accepted), Arles, Mar. 2015.

[2] W. Chen, A. Plancoulaine, N. Férey, D. Touraine, J. Nelson, and P. Bourdot. 6DoF navigation in virtual worlds: comparison of joystick-based and head-controlled paradigms. In *19th ACM Symp on Virtual Reality Soft and Tech (VRST)*, pages 111–114, New York, 2013. ACM.

[3] D. Felinto and M. Pan. *Game Development with Blender*. Cengage Learning, 1st ed., 2013.

[4] H. Li. Création et contrôle de l'avatar dans la plateforme BlenderCave. Master's thesis, Université Paris-Sud, 2014.

[5] D. Poirier-Quinot and B. F. Katz. CAVE-based virtual prototyping of an audio radiogoniometer: Ecological validity assessment. In *20th Intl Conf on Auditory Display (ICAD)*, pages 1–8, NYC, June 2014.

[6] D. Poirier-Quinot, D. Touraine, and B. F. Katz. BlenderCAVE: A flexible open source authoring tool dedicated to multimodal virtual reality. In *5th Joint Virtual Reality Conf (JVRC)*, pages 19–22, Orsay, 2013.

[7] D. Poirier-Quinot, D. Touraine, and B. F. Katz. BlenderCAVE: A multimodal scene graph editor for virtual reality. In *19th Intl Conf on Auditory Display (ICAD)*, pages 223–230, Lodz, Oct 2013.

[8] M. Rébillat, E. Corteel, and B. Katz. SMART-I2 "Spatial Multi-user Audio-visual Real-Time Interactive Interface". In *125th Conv of the Audio Eng Soc*, pages 7609_1–16, 2008.

[1] http://BlenderVR.limsi.fr

[2] http://www.blender.org

[3] http://github.com/BlenderVR

[4] http://www.limsi.fr/venise/EVEsystem