

A Framework for Immersive VR and Full-Body Avatar Interaction

Carlo Camporesi*

Marcelo Kallmann†

School of Engineering
University of California, Merced

ABSTRACT

We present a new software framework for the development of immersive collaborative virtual reality applications with emphasis on full-body interaction with virtual humans. The framework is designed to be scalable and adaptable to different hardware configurations, using either commodity or more expensive hardware setups. High level programming abstractions handle object replication and communication channels for configurations based on multiple rendering nodes. The system provides tools for full-body character animation, real-time full-body tracking and retargeting to virtual humans, including motion reconstruction from a small set of sensors. Such capabilities allow new full-body interaction metaphors to be developed in fixed or user-perspective stereo visualization. The framework is scalable, customizable, cross-platform and it is only based on tools with source code available.

Index Terms: I.3.2 [Computing Methodologies]: Computer Graphics—Distributed/network graphics.

1 INTRODUCTION

Collaborative and immersive Virtual Reality is becoming increasingly popular. Current solutions are however designed around specific hardware, designed to solve specific problems, and/or designed as a commercial solution. In addition, a gap can be often observed between the available capabilities and what is offered by state of the art graphics engines in terms of qualitative rendering, performance, or high-level programming tools available to the developer [5].

Immersive VR systems started to appear in the 90s and have defined what can be called cluster-based virtual reality. Such systems can be subdivided in two main categories: Client-Server (CS) or Master-Slave (MS). Our system follows a MS architecture, similarly to Vrui [7] and Avango [8], which are VR toolkits similar to our framework. They both integrate a scene graph abstraction but they do not well provide character animation tools and integration with high-end modeling packages. Jacobson et al. [4] proposed CaveUT based on Unreal Tournament 2004; Juarez et al. [5] proposed a distributed environment based on CryEngine 2; however both engines are closed solutions not providing cross-platform portability or tools for networked collaboration. More recently, MiddleVR [3] was released as a plug-in for Unity 3D. It provides support for generic hardware configurations however it is also a closed commercial solution. Tele-immersive systems with focus on high-quality user appearance have also been proposed [10, 9]. These systems are specialized for their applications and are also not suitable for multi-user connections due the high bandwidth required.

We present a scalable system for generic hardware configurations that provides integration with high-end graphics tools (OGRE

rendering engine) and advanced solutions enabling real-time full-body interaction with simulated virtual humans (see Figure 1).

2 SYSTEM ARCHITECTURE

The framework is implemented in C++ and its main entry point is a derivable core class. Depending on the functionality required by the application, different modules instantiated as managers can be extended (see Figure 2).



Figure 1: Example applications with the framework in a low-cost distributed multi-tile wall.

The system core extends the OGRE rendering cycle. The application core is a monolithic entity that orchestrates the requested managers, maintaining the shared states, handling the user's interaction events and providing to the developer a high-level abstraction of the system. The rendering cycle follows the standard update-render-swap mechanism. In case of clustered rendering, the master instance drives the rendering by enforcing a synchronized swap step of the video buffers. The camera frame module, integrated in the core, is responsible for creating the physical rendering window array and for performing generic virtual camera manipulations. In order to support a wide range of configurations the system employs a *virtual screen, window and viewport* model. The camera frame supports multi tiled clustered displays, multi projection systems and 3D TVs with passive and active stereo.

The network communication between instances has been designed to be extensible as needed and to accommodate multiple communication engines that can be selected according to specific needs. Two default engines are provided with the framework (both deriving from the network library RakNet). The transmission protocol used is UDP datagram extended with a communication reliability layer with optional packet ordering and sequencing features, data compression, bandwidth saving mechanism and security layer. The first engine supports the internal cluster communication (master-slave interactions) and implements two communication layers. The first layer is based on a low-level message passing mechanism (sliding window process with mutual waits), driving the internal nodes synchronization, core event messages and maintenance of the internal core's shared state. The second layer of communication has been designed to simplify the creation, maintenance

*e-mail: ccamporesi@ucmerced.edu

†e-mail: mkallmann@ucmerced.edu

and update of networked objects through *communication channels*. The concept extends the distributed object replica pattern. The second engine provides general purpose communication channels between collaborative application instances (master-master interaction designed for WAN connections).

The device manager provides a simple mechanism to allocate interaction devices. When an input device is requested a virtual device is created and its device-dependent messages are translated into a virtualized higher level format. The framework natively supports mice, keyboards and joysticks. A client-server extension has been developed to retrieve information from platform-dependent devices, such as: Vicon Blade's motion capture streaming, Opti-Track, 5DT data-gloves, 3D-Connexion mice, etc. Devices can be assigned as camera manipulators, head tracking entities or bound to virtual pointers to interact with the virtual objects and the 3D GUI system. The 3D GUI manager includes primitives for distributed interactive widgets with user perspective interaction. The system supports specialized widgets like buttons, labels, spin-boxes, sliders and video planes as well as advanced pre-built panels designed to handle the motion animation tasks. The coexistence of widgets and pointers grants to the user a full scale calibrated 3D user-perspective immersive interaction model.

The character animation and motion reconstruction modules are encapsulated in a character class that handles the loading of high-resolution skinned models. The animation module includes tools to compute and visualize on-line information from motions being tracked, as for example: angles between limbs, end-effector trajectories, and joint range of motion.

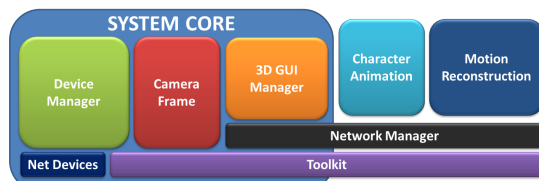


Figure 2: Overall system architecture.

The motion reconstruction module allows real-time full-body tracking and retargeting to virtual characters of varied dimensions. The motion tracking process can be performed with full-body tracking (through Microsoft Kinect, inertial sensors or Vicon cameras). A motion retargeting calibration and real-time mapping algorithm from a reduced set of sensors is available based on real-time Inverse Kinematics. Finger motions can also be reconstructed from data-gloves. The reconstruction algorithms follow the solutions discussed by Camporesi et al. [1].

3 FULL-BODY INTERFACES ENABLED BY THE SYSTEM

Our framework has been successfully employed on a number of research projects exploring new immersive full-body interfaces and interactions with avatars and virtual characters. Examples are given below.

Motion Modeling This application focuses on training scenarios where the user (expert in the training subject) is able to demonstrate to a virtual human how to perform parameterized motions. Later, the virtual human is able to reproduce the motions autonomously in interactive training sessions. The system includes new algorithmic solutions for motion parameterization and fast motion mapping to the avatar from a reduced marker set.

Physical Therapy The presented framework is being employed to explore new VR approaches to improve the delivery of upper-body physical therapy. The application allows therapists to create new exercises by direct demonstration, and it then provides automatic interactive exercise delivery and monitoring with an au-

tonomous virtual therapist. The application also supports networked collaboration. See Figure 1-right.

Interactive Motion Visualization A full-scale visualization tool has also been implemented for inspection of the results computed by a full-body motion planning algorithm [2]. Motions are synthesized from a motion graph algorithm in an environment with obstacles, and then integrated with a motion planner designed to compute upper-body motions for given tasks. Powerful interactions are enabled with the possible immersive manipulation of the environment, allowing users to closely evaluate the capabilities and limitations of the involved planning algorithms.

Additional Projects Additional applications supported by the framework include: visualization application for high-resolution laser scanner data and cloud of points, 3D archeological fieldwork video player, visualization of archeological environments (see Figure 1-left) with interactive content navigation and display [6], and tools for virtual teaching and virtual presentations.

4 RESULTS AND CONCLUSION

Our framework has been fine-tuned, tested and evaluated in different physical configurations, from simple desktop computers to distributed clustered environments. The primary test platform for our immersive virtual reality setup is a six-tile Powerwall, and integrated with a 10-camera Vicon motion capture system. The Powerwall is retro-projected (4.56m by 2.25m) and consists of seven commodity Linux-based rendering nodes (Pentium Q9550 2.83GHz GeForce GTX 280 4Gb RAM), twelve projectors with circular passive polarization filters (1024x768px resolution each), an external windows device server, and a Gigabit Ethernet. We also performed experiments simulating CAVE-like environments (from the standard 4-walls configurations to more customized multi-tile screen compositions), 3D TVs and using different motion tracking devices.

ACKNOWLEDGEMENTS

This work was partially supported by NSF award IIS-0915665 and by CITRIS seed project 128.

REFERENCES

- [1] C. Camporesi, Y. Huang, and M. Kallmann. Interactive motion modeling and parameterization by direct demonstration. In *Proc. of the 10th International Conference on Intelligent Virtual Agents (IVA)*, 2010.
- [2] Y. Huang, M. Mahmudi, and M. Kallmann. Planning humanlike actions in blending spaces. In *Proceedings of the conference on Intelligent Robots and Systems (IROS)*, 2011.
- [3] i'm in VR. Middle VR: Unity VR Plug-In. <http://www.imin.fr/middlevr/>.
- [4] J. Jacobson and M. Lewis. Game engine virtual reality with caveat. *Computer*, 38(4):79–82, april 2005.
- [5] A. Juarez, B. Schonenberg, and C. Bartneck. Implementing a low-cost cave system using the cryengine2. *Entertainment Computing*, 1(3-4):157–164, 2010.
- [6] S. Kenderdine, M. Forte, and C. Camporesi. Rhizome of western han: an omnispacial theatre for archaeology. In *Proceedings of the 39th International Conference CAA*, Amsterdam, 2011. Pallas Publications.
- [7] O. Kreylos. Environment-independent vr development. In *Proc. of the 4th International Symposium on Advances in Visual Computing, ISVC '08*, pages 901–912, Berlin, Heidelberg, 2008. Springer-Verlag.
- [8] R. Kuck, J. Wind, K. Riege, and M. Bogen. Improving the avango vr/ar framework: Lessons learned. In *Workshop Virtuelle und Erweiterte Realität*, pages 209–220, 2008.
- [9] S. min Rhee, R. Ziegler, J. Park, M. Naef, M. Gross, and S. Member. Low-cost telepresence for collaborative virtual environments. in *IEEE Trans Vis Comput Graph*, 13:156–166, 2007.
- [10] Z. Yang, B. Yu, W. Wu, R. Diankov, and R. Bajscy. Collaborative dancing in tele-immersive environment. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA '06, pages 723–726, New York, NY, USA, 2006. ACM.