

A Framework for Immersive VR and Full-Body Avatar Interaction

Carlo Camporesi and Marcelo Kallmann
{ccamporesi, mkallmann}@ucmerced.edu



Abstract

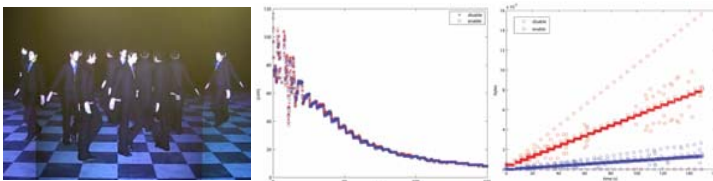
We present a new software framework for the development of distributed immersive collaborative virtual reality applications with emphasis on full-body interaction with virtual humans.

- scalable to multi tiles, customizable and cross-platform.
- automatically distributed object replication.
- automatic state registration and maintenance.
- full-body synchronized character animation tools.
- real-time full-body tracking and retargeting.
- based on open source tools or with source code available.



The framework in a low-cost distributed multi-tile wall: Immersive object manipulation and 3D GUI interaction, virtual heritage reconstructions [2] and motion modeling [1].

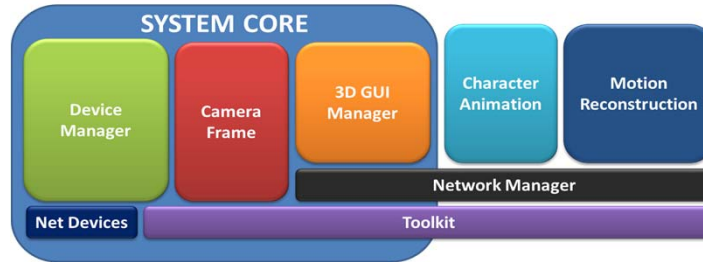
Evaluations



Hardware setup: Seven Linux-based rendering node Powerwall (P-Q9550 2.83GHz, GeForce GTX 280, 4Gb RAM), twelve projectors and an external windows device server with a 10-camera Vicon motion capture system.

Comparison between applications with the bandwidth saving mechanism enabled (blue markers) and disabled (red markers) and preservation of the frame rate.
Left: test application: high-res characters are loaded every 5 seconds. Center: frame rate performances compared to non-distributed version of the system. Right: bytes transmitted

System Architecture



Overall system architecture: framework modules organization

• System Core

Extends the OGRE rendering engine cycle. The application core is a monolithic entity orchestrating 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 for fast application development.

• Character Animation

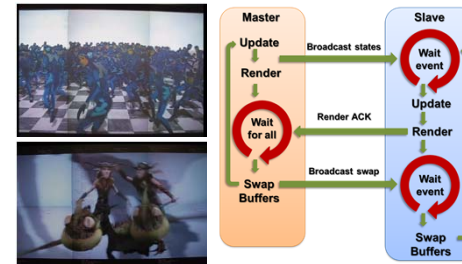
Handles the loading of high resolution skinned models. Supports distributed key-frame animation interpolations and distributed joint-values over-time animations. The character networked synchronization is bandwidth efficient and supports automatic determination of minimal updates.

• Motion Reconstruction

Allows real-time full-body tracking and retargeting to virtual characters of varied dimensions 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 inverse kinematics [1].

• Camera Frame

Creates the physical rendering window array (supporting user-perspective stereo rendering) and performs generic virtual camera manipulations. The system employs the "screen, window and viewport" model supporting many system configurations (CAVEs, Tiled Walls or simple 3D TVs).



Right: Rendering cycle synchronization. Left: Sync tests with high-res video planes and key-framed skeletal animation crowd.

• Network Manager

The framework provides an internal cluster communication engine (master-slave interactions with a sliding window) for the cluster rendering/state synchronization and a general purpose communication channels (master-master interaction). Customized engines can be added.

The framework is implemented in C++ and its main entry point is a derivable core class. Depending on the functionality required, different modules, instantiated as managers, can be requested.

The core is highly flexible and application's instances can also be customized from a unified configuration file.

• Device Manager and the 3D GUI

This manager handles interaction devices. Virtual devices are created on request and virtualized to a higher level format. A client-server extension handles platform-dependent devices. The 3D GUI manager includes primitives for distributed interactive widgets.

Full-Body Interfaces Enabled by the Framework

Our framework has been successfully employed on a number of research projects for more than three years [1][2].

• Motion Modeling

User demonstrates to a virtual human how to perform motions. During the training phase the virtual human reproduces the motions in interactive training sessions with apprentice users learning the training subject [1]. The system includes new algorithmic solutions for motion parameterization [3].



• Physical Therapy

Allows therapists to create new upper-body exercises by direct demonstration, and then the autonomous virtual therapist provides automatic interactive exercise delivery and monitoring to users. The application also supports networked collaboration.



• Interactive Motion Visualization

Provides full-scale visualization for inspection of synthesized full-body motions. Inspected motions have been synthesized from a motion graph algorithm with obstacles, and then integrated with a multi modal motion planner.

References

- [1] C. Camporesi, Y. Huang and M. Kallmann, *Interactive Motion Modeling and Parameterization by Direct Demonstration*, IVA 2010.
- [2] S. Kenderdine, M. Forte, and C. Camporesi, *Rhizome of western han: an omnispatial theatre for archaeology*. CAA, 2011
- [3] Y. Huang and M. Kallmann, *Motion Parameterization with Inverse Blending*, MIG 2010.

Acknowledgments: this work was partially supported by NSF award IIS-0915665 and by CITRIS seed project #128.