A Framework for Haptic Rendering of Large-Scale Virtual Environments

Mashhuda Glencross* and Roger Hubbold
Advanced Interfaces Group, Department of Computer Science,
University of Manchester, UK

1 Introduction

The CoHap3D framework is a prototype middle-ware system for haptically enabling 3D VR applications. It is aimed at haptic rendering of large-scale models in a single or multi-user context.

CoHap3D incorporates a 3D graphics system, MAVERIK^{++}, based on our own GNU/MAVERIK library but modernised to exploit recent developments in graphics hardware. Further, a haptics framework is implemented which makes use of spatial partitioning techniques to manage the complexity of haptic rendering through a client-side haptic cache.

The system currently supports the FCS HapticMASTER device but can easily be extended to incorporate other haptic devices.

The only other similar framework currently under development is Stanford University and the Universita' di Siena's CHAI 3D[1]. GHOST provides a lower-level framework than CoHap3D and commercial systems such as Reachin's API [Reachin 2004] and Novint's e-Touch [etouch 2004] toolkit do not provide sufficient support for building large-scale VR applications.

In this sketch we focus primarily on the support for haptic rendering of large-scale virtual environments in the CoHap3D framework.

2 The CoHap3D System

To convey a believable sense of touch, a haptic response must be rendered at a speed of 1KHz [Burdea 1996], or better. Hence, haptic devices employ a client-server architecture. A user's client connects to a haptic server whose primary function is haptic rendering.

The CoHap3D framework provides client-side functionality for application developers. On the client side, the application calls into the CoHap3D framework which in turn calls into the haptic API. This communicates via CORBA with a haptic server PC running the VxWorks real-time operating system. The haptic server operates a fast (2.5KHz) haptic rendering loop giving an impression of very solid contact. CoHap3D maintains consistent state between the client and the haptic server. This is achieved by consolidating, into single objects, the graphical and haptic representations of geometric primitives such that any change in the state of a graphical primitive automatically propagates to the haptic counterpart, freeing the application developer from syncronization concerns.

In the case of large-scale model rendering, the situation is a bit more complex; the fast haptic rendering loop, imposes severe limitations on the number of haptically enabled primitives that can be handled by the server.

The CoHap3D framework provides a solution to this problem by using spatial partitioning techniques conventionally used for culling to a view frustrum. The original implementation of GNU/MAVERIK contained extensive support for rendering large-scale models [Hubbold et al. 2001], including hierarchies of axis aligned bounding boxes (AABB), voxel-based partitioning of space [Clark 1976; Samet 1989], and the use of landmarks.

In the CoHap3D system these techniques have been exploited to cull primitives on the client side to a haptic frustrum to identify those in the vicinity of the haptic workspace. A list of potential candidates which may need to be haptically enabled is identified and cached, proximity tests between these and the user's position are used to further narrow the list to be activated in the haptic server. Cached primitives are added and removed as the user navigates through the environment. Navigation is either via a conventional 2DOF mouse, a 3DOF mouse, or the HapticMASTER itself used as a rate controlled joystick.

An application has been implemented which uses the client-side haptic cache to haptically enable geometric primitives (coloured dark blue) in the vicinity of the user. The application is able to cope with models containing tens to hundreds of thousands of geometric primitives. The following two screen-shots show a user haptically interacting with a model containing 24,962 distinct primitives running on a Dell Latitude laptop computer with a 1.2GHz processor, 500MB of memory, and a GeForce2Go graphics chip.

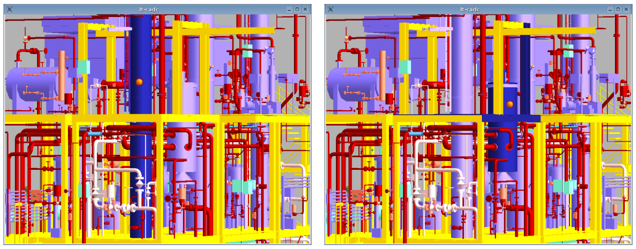

Figure 1: Screen shots showing a large-scale haptic rendering application built using CoHap3D.

This work is sponsored by the UK Engineering and Physical Sciences Research Council, under grant GR/S23087/01.

Further information about CoHap3D and other research projects at Manchester can be found at http://aig.cs.man.ac.uk/research/.

References

BURDEA, G. C. 1996. *Force and Touch Feedback for Virtual Reality*. John Wiley and Sons, Inc.

CLARK, J. H. 1976. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM 19*, 10 (Oct.), 547–554.

ETOUCH, 2004. etouch home page. Online document. ⟨http://www.etouch3d.org/index.html⟩.

HUBBOLD, R., COOK, J., KEATES, M., GIBSON, S., HOWARD, T., MURTA, A., WEST, A., AND PETTIFER, S. 2001. GNU/MAVERIK : A micro-kernel for large-scale virtual environments. *Presence: Teloperators and Virtual Environments*, 10 (February), 22–34. ISSN 1054-7460.

REACHIN, 2004. Reachin API online reference manual. Online document. ⟨http://www.reachin.se/support/documents/referenceManual/⟩.

SAMET, H. 1989. *Spatial Data Structures: Quadtree, Octrees and Other Hierarchical Methods*. Addison Wesley.

*e-mail: khotem@cs.man.ac.uk

[1] http://www.chai3d.org/