A Haptic Toolkit for the Development of Immersive and Web-Enabled Games

E. Ruffaldi, A. Frisoli, M. Bergamasco, C. Gottlieb, F. Tecchia
PERCRO
Scuola Superiore Sant'Anna

ABSTRACT

The creation of applications of Virtual Reality enabled with Haptic interaction and dynamic simulation requires usually to cover many implementation details that increase the development time and the effectiveness of the application itself. This work presents one game application that has been developed using a Haptic toolkit for the rapid application development, that integrates 3D graphics, haptic feedback and dynamic simulation. The resulting application can be easily deployed on the Web directly to the final user.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Haptic I/O; H.3.5 [**Online Information Services**]: Web-based services

General Terms

Algorithms

Keywords

Haptics, Web, Virtual Enviroments, game

1. INTRODUCTION

Haptic interfaces have been around for more than a decade. However, only recently a substantial decrease in price has allowed a strong diffusion of these devices into many research centers and industries. At the same time performance improvements of graphics hardware has made possible visualizing complex realistic environments. These advances of the hardware require additional efforts in the software, and there is the opportunity of creating haptic enabled multimodal systems.

The objective of this work is the presentation of a tool for haptic interaction on the Web. The purpose of this tool is to stimulate the spread of haptic applications by improving their development and deployment, in particular in the distribution of haptic enabled applications. Our goal is to improve the teaching and experimentation of haptics among students, the rapid prototyping of applications and the construction of experiments for the validation of the perceptual aspects. The motiviation is that there are many applications in which haptics can be applied fields such as virtual prototyping, virtual museums and training.

The integration of haptics with the Web has been addressed by integrating a VRML browser with haptic rendering with the addition of extension nodes [7, 6]. One of the most promising solutions is H3D by SenseGraphics that provides an X3D implementation based on Sensable's OpenHaptics. The first limitation of these systems is the support of a specific family of devices and interaction modalities. The other aspect that we want to point out is the dependency on the VRML format. This is clearly a choice that favours compatibility and standardization although there are no standards for haptic nodes. The use of VRML decides the interaction mode and the logical structure of the application.

In HapticWeb we decided to provide more flexibility to the developer in the choice of the system organization because the experimentation of interaction paradigms and application structures is among our objectives. At the same time we are committed to provide interoperability and stupport for the Web standards with the aim of integrating the haptic applications. The rest of the paper is structured as follows: first a description of the organization of HapticWeb is presented and then a pool game application that have been developed using this tool.

2. ARCHITECTURE

HapticWeb is a script based framework for the development of haptic enabled application that are almost independent from the specific haptic interface [1] and it provides enough flexibility for its extension with additional modules, a feature that is fundamental for the creation of complete multimodal applicationss. The current HapticWeb system is based on the Virtual Reality engine eXtreme Virtual Reality (XVR) [3] constitued by a fast and simple bytecode virtual machine, a set of core 3D graphics features and a deployment mechanism that allows an easy distribution of the application on the Web. HapticWeb provides new modules for haptic interfaces, haptic rendering and dynamic simulation not available natively in XVR, and also a set of higher level classes that simplify the development of multimodal applications. Figure 1 shows the HapticWeb architecture

VRST'06, November 1–3, 2006, Limassol, Cyprus.
Copyright 2006 ACM 1-59593-321-2/06/0011 ...$5.00.

[1] Being based on CHAI3D, HapticWeb supports most commercial kinesthetic devices based on impedance control

with the core XVR modules and the modules provided by HapticWeb [2].

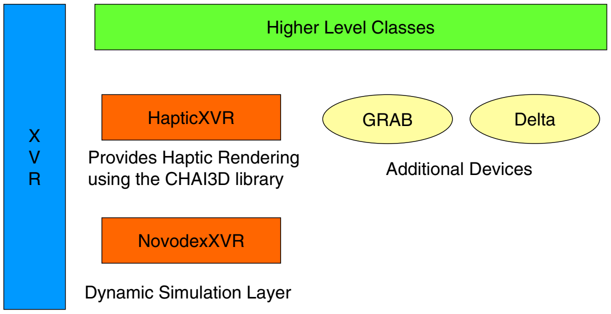


Figure 1: The architectural view of HapticWeb

The HapticWeb application is described by a scripting language specialized for 3D graphics that is able to control the different modalities through an extensive object model, and at the same time it allows the low level control of the graphics through OpenGL commands. After the download and the initialization phase the developer has complete control over the graphics and haptic rendering, and he is required to define the application logic and the interaction between the different modalities. When the developer has prepared the script program and the associated multimedia resources, he publishes it on the Web inside the Web page, along with archives containing the resources. The application is executed inside the Web Browser using a plugin for 3D graphics that loads the program, the associated 3D models, and executes the application. The HapticWeb program is associated with a specific version of the runtime engine and the plugin automatically downloads the requested version. With this approach we separate the update of the application, done by downloading the code from the Web at each execution, and the update of the HapticWeb runtime automatically performed by the plugin.

2.1 XVR

The HapticWeb application is deployed on a Web site as a compiled program and a set of multimodal resources that can be downloaded from the network and made accessible from a Web page. The execution of the program takes place inside a Web Browser's plugin that provides the integration with the Web page and the network. A virtual machine evaluates the program in the form of a bytecode representation. The bytecode representation for the distribution of Web applications has been successful in commercial systems like Java and Macromedia Flash, because of the compactness of the representation, support for multiple platforms and security. Additionally the bytecode adopted for this project has been tailored for 3D graphics applications.

The execution environment, the Web Browser's plugin, and the virtual machine for this system are provided by the XVR engine developed by PERCRO laboratory and presented in [3]. The XVR platform has been used in many Virtual Reality projects running both on the Web or inside highly immersive installations, and it is also used in a Virtual Reality course.

A HapticWeb program is written using the XVR scripting language, an object oriented scripting language specialized for 3D graphics, with a C-like syntax that resembles current shading languages. Its specialization is focused on the efficient management of vectors, such as support for the swizzle operator.

XVR applications are organized around callback functions invoked during specific events and system loops. In general it is possible to describe a multimodal application as a multirate program, with a common logic that coordinates the different modalities. The logical loops of the XVR platform are shown in Figure 2 but for making the structure of the application simpler only two of them are explictly associated with callbacks: the graphics loop and a generic timer loop, the first running at the display refresh rate and the other at 1KHz.

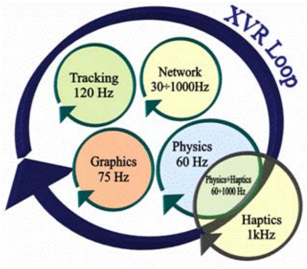


Figure 2: This diagram shows the loops of a typical multimodal system withing XVR

The XVR runtime engine provides support for 3D graphics and audio spatialization exposed as an object model accessible to the developer in the program. The developer has low level control of the graphics through OpenGL commands and at the same time the visualization of complex, multi-material and animated models generated with standard 3D modellers. The low level access provides flexibility in the generation of specific 3D elements or effects whereas the high level access provides the standard scene graph approach for the visualization of 3D graphics.

2.2 The CHAI Haptic library

The XVR's object model has been extended to support haptic feedback providing support for devices, haptic rendering of objects,and haptic effects, expanding in this way the range of possible applications of the XVR platform. The haptic functionalities of the HapticWeb platform are provided by the CHAI 3D haptic library [4], an Open Source effort of Stanford University for multiple haptic devices and multiple platforms. We have chosen this library because it is device independent and it provides different choices for the implementation of the haptic loop. CHAI uses a haptic scene graph for organizing objects and points of contact of the devices. In this project we have exposed most of CHAI's features to the scripting system and extended them for haptic feedback realism and expressiveness.

2.3 Device Access

The support of multiple haptic devices is one of the fundamental requirements for the spreading of a haptic application framework, and HapticWeb takes advantage of the variety of haptic devices supported by the CHAI3D library. The overhead for such flexibility is limited and can be measured as an additional 150K of data downloaded for supporting all of them. In a single device configuration we try to figure out which device is currently active and initialize it. Eventually, if there is no haptic device attached, a device simulated using the mouse is used. HapticWeb provides the possibility of initializing different devices by using a device URI. A device URI specifies the type of device, its identification if there are many of them and optional parameters expressed using the query part of the URI. For example a PHANTOM can be accessed using “phantom://default?mode=direct” where the mode parameter specifies use of Direct I/O if available. In the case of a remote device we can write “remote://145.22.33.44/device0”. In general we suggest using the automatic device configuration because it reaches the maximum audience but in some cases a specific one could be required. Multiple devices can be instantiated as well, and each can be associated with a tool for the force rendering of the point of contact. In the case of the GRAB device [1] with two arms, each arm is identified by a different URL and requires separate haptic rendering: “ehap://grab/left” and “ehap://grab/right”.

One of the problems in the portability of haptic applications is the difference between the devices in terms of workspace and force feedback limits. We address the problem by providing to the developer detailed information about the device, a feature that is not available in most of libraries and also we avoid any workspace scaling that affect the realism of force feedback.

2.4 Expressing Haptics

The haptic feedback capabilities presented so far are relative only to the standard object touching interaction that can be obtained using proxy-based algorithms, but a haptic application sometimes requires the generation of force effects for notification of events to the user or force fields for constraints and guidance. For example a needle insertion simulation could provide a force field as a haptic hint for the task. HapticWeb provides a set of force effects that can be applied as superimposed or as alternatives to the surface feedback. Each effect can be enabled explicitly or it is associated with a time duration useful for triggered effect. At the same time there is an activation volume in which the effect is active represented by a sphere. Finally the force exerted by the effects can be expressed locally or globally to the haptic point of contact.

The effects that we provide can be categorized by generic force fields and constraints. The first group contains simple effects like spring, implicit sphere, virtual plane, friction or constant force associated with a certain bounding volume. However the constraint effects are described using a set of points, lines and triangles that specify attraction points for the haptic point of contact, with different levels of constraint strength. The resultant force from the haptic loop is obtained by exclusively choosing between the nearest active constraint and the surface feedback, and superimposing to that the other forces caused by the other active force field effects.

The use of constraints, in particular of spline based curves, can be used also for volume visualization applications as in [5], that discuss the integration of constraints with proxy algorithms.

The geometry used for the haptic feedback is typically obtained from one precomputed model, but there are cases in which we need to construct it dynamically or we have existing code that visualizes some geometry using OpenGL commands. For the above reasons HapticWeb provides a geometry capture feature that allows construction of the geometry used for the haptic rendering from the OpenGL rendering. The geometry inside the view frustum that is sent to OpenGL inside the capture region is used for updating the haptic mesh. The other application of this technique is for the specification of the haptic constraints discussed above. Every point or line sent to OpenGL inside the effect capture region is transformed into a constraint primitive[2].

3. THE POOL DEMO APPLICATION

A complete example of application using HapticWeb is the Haptic Pool, that allows to play Billiards using a haptic interface. This example integrates the dynamic simulation of the pool table with the haptic feedback using the HapticWeb framework described above. The haptic interface is used for impressing force and direction to the balls, and also for changing the point of view of the player, using the direct rendering of the forces. Figure 3 shows the application while playing with the GRAB device. The application is enhanced with audio feedback to provide the sound of collisions between the balls with the cushions and other balls. The user decides the hit direction through the haptic interface; then by pressing a button on the device, a virtual sliding is implemented that constraints the cue to move only forward and backward along a line aligned with the hit direction and through a point $\mathbf{p}$ of the ball, that represents the hit point. Basically the force-feedback is computed as an impulse force assumed proportional to the hitting velocity, $F_{hit} = kv_{hit}$. If T is the sampling time, an impulse force $I_{hit} = F_{hit}T$, is then applied to the ball at the position $\mathbf{p}$ where the cue hits the ball, the linear and rolling initial conditions of the dynamics of the ball are given by:

$$\begin{cases} m\mathbf{v_{in}} = I_{hit} \\ I\boldsymbol{\omega_{in}} = \mathbf{p} \wedge I_{hit} \end{cases} \tag{1}$$

The hit point $\mathbf{p}$ can be changed by the user through the arrow keys to implement underspinning effects, see for instance the green point in figure 5. Billiard cloth is implemented through static μ_s and dynamic friction μ_d properties, and with an additional constant force term $F_{el} = k_2 mg$ proportional to the ball weight, that models the dissipation of energy due to the elasticity of the cloth under the weight of the balls.

Then the free dynamics of the ball is computed to determine the evolution of position of the ball over time, until collisions either with other balls or cushions happen. In static conditions we have, indicating with R the ball radius, and by considering the moment equilibrium equation at the contact point

$$\begin{cases} v = \omega R \\ I\frac{d\omega}{dt} = -F_{el}R \end{cases} \tag{2}$$

while in dynamic conditions, with sliding occurring between the ball and the cloth

$$\begin{cases} m\frac{dv}{dt} = -\mu_d mg - F_{el} \\ I\frac{d\omega}{dt} = -F_{el}R \end{cases} \tag{3}$$

[2] The geometry capture for a haptic mesh could be performed inside the haptic world semaphore, but this could completly stall the haptic rendering, for this reason is better first to disable the object from the haptic world and then perform the geometry capture operation.

The collisions are modeled with simple geometry reflection rules and conservation of momentum, but considering a restitution coefficient that is a function of the material of the colliding objects, modeling dissipating phenomena in the collision. Cushions are modelled with suitable height and contact radius, in order to predict the correct collision behavior. All the dynamics is implemented through the Novodex dynamic simulation engine.

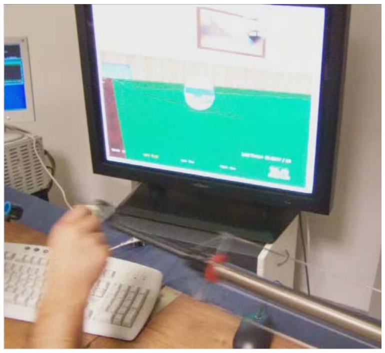


Figure 3: Example of the Haptic Pool application in which the GRAB device is being used

In figures 4 and 5 the possibilities offered by the application are shown, like real time collision detection capability and dynamics with modeling of rolling of balls, and possiblity of applying spinning effects when hitting the balls, by varying the point of application of force **p**.

An on-line demo of the haptic pool game is available on https://www.enactivenetwork.org/MMM/page3.html. The demo to work properly requires a MS Internet Explorer browser and an haptic interface supported by HapticWeb. The same web-page shows how the hapticweb 3D interactive technology can be easily integrated within a standard html web-page.

4. REFERENCES

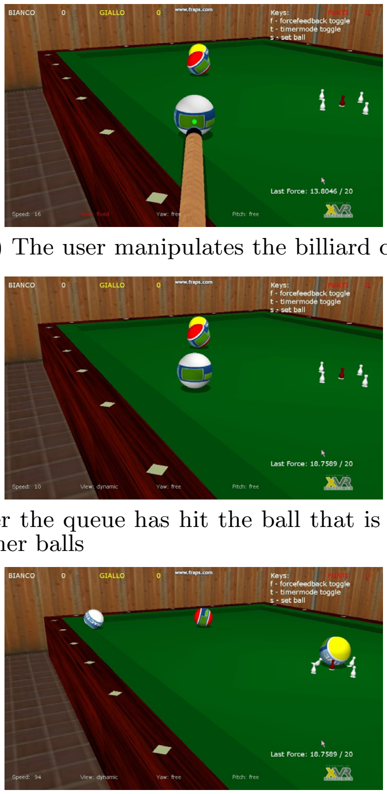
[1] C. Avizzano, T. Gutierrez, S. Casado, B. Gallagher, M. Magennis, J. Wood, K. Gladstone, H. Graupp, J. A. Munoz, E. F. C. Arias, and F. Slevin. Grab: Computer graphics access for blind people through a haptic and audio virtual environment. In *Proceedings of Eurohaptics 2003*, July 2003.

[2] M. Bergamasco, E. Ruffaldi, and C. A. Avizzano. Enactive and internet applications: A first prototype of enactive application that interoperates haptic devices and the world wide web, 2006.

[3] M. Carrozzino, F.Tecchia, S.Bacinelli, and M.Bergamasco. Lowering the development time of multimodal interactive application: The real-life experience of the XVR project. In *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 2005.

[4] F. Conti, F. Barbagli, D. Morris, and C. Sewell. CHAI, an open-source library for the rapid development of haptic scenes. In *Hands-On-Demo at the World Haptics Conference*, 2005.

[5] K. Lundin, B. Gudmundsson, and A. Ynnerman. General proxy-based haptics for volume visualization. In *WHC*, pages 557–560, 2005.

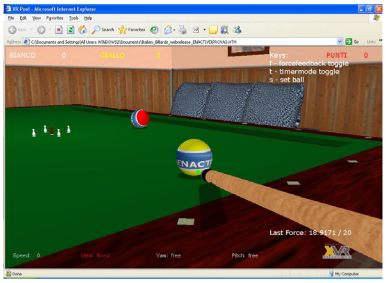


(a) The user manipulates the billiard cue

(b) Soon after the queue has hit the ball that is travelling towards the other balls

(c) The balls hit cushions and birilli

Figure 4: A sequence of snapshots of the pooling demo application



(a) Underspinning hit of the ball

Figure 5: Possibility of adding spinning effects while hitting the ball

[6] M. McLaughlin, G. Sukhatme, C. Shahabi, J. Hespanha, A. Ortega, and G. Medioni. The Haptic Museum. *Proceedings of the EVA 2000 Conference on Electronic Imaging and the Visual Arts*, 2000.

[7] M. K. O'Malley and S. Hughes. Simplified authoring of 3D haptic content for the World Wide Web. In *HAPTICS*, pages 428–429, 2003.