Extending DIMPLE: a rigid body haptic simulator for interactive control of sound.

Stephen Sinclair Marcelo M. Wanderley

Input Devices and Music Interaction Laboratory, McGill University
Centre for Interdisciplinary Research in Music Media and Technology

E-mail: sinclair@music.mcgill.ca, marcelo.wanderley@mcgill.ca

Abstract

We previously introduced DIMPLE, a software environment allowing the run-time creation of a physically dynamic, haptically-enabled virtual scene using the Open Sound Control (OSC) protocol [18]. Object properties could be requested over OSC to allow modulation of parameters for sound synthesis or visualization. Examples were given of PureData patches that controlled such scenes, making haptic virtual musical instruments (VMI) accessible to a visual programming environment. However, scenes were limited to simple 3D objects and haptic interaction was limited to pushing on objects and their constraints. Here we present various recent developments to enhance DIMPLE for more intricate haptic sensations and to allow a wider variety of user interaction techniques.

1 Introduction

DIMPLE is composed of a multithreaded interface between the CHAI haptic C++ framework [5] and the Open Dynamics Engine [20], with procedure calls exposed through an OSC interface for creating objects and accessing their properties.

Since the physics and haptics systems run asynchronously, the update interval of the physics simulation is independent of the haptic resolution. While haptic rendering occurs at the required 1 KHz interval necessary to present stable and stiff surfaces [12], forces accumulated on a rigid body by the haptic proxy are applied at the next physics timestep, which is configurable from 30 to 1000 Hz (usually 100 Hz).

Currently, it is possible to create simple objects such as prisms and spheres, as well as to combine these shapes into compound objects sharing a single rigid body. Constraint-based relationships between objects or between an object and the global coordinate system can be specified, allowing hinges or springs (for example) to be specified.

2 Improvements

While pushing objects around and using their dynamic behaviour to modulate parameters can be interesting, we acknowledge that much of the subtlety achieved in musical interaction occurs at the sub-millimeter level, with friction, textured surfaces, and tactile vibration. For DIMPLE to be musically interesting, it is necessary to address deficiencies in these areas. We report here some recent progress towards this goal.

2.1 Vibro-tactile feedback

A feature lacking previously in DIMPLE was good feedback from the audio synthesis to the haptic rendering system. It has been shown that vibro-tactile feedback is important in making controllers feel "alive" and interesting to performers [11], and also that it can be critical to performance of unpredictable and non-linear instrument characteristics [4]. Thus it would be a pity to ignore this important property of virtual controllers.

Combined haptic-audio physical modelling systems such as CORDIS-ANIMA [3] make use of a single physical model, rendered synchronously, to produce both haptic rendering and audio synthesis that are tightly integrated. Haptic frequencies present in the vibration of objects at the audio rate are also felt in the haptic feedback due to this combined approach. DIMPLE, in contrast, has taken an approach emphasizing the use of haptics for manipulation of objects, but producing audio feedback in a separated synthesis engine (PureData or other) that is decoupled from the haptic/physics simulation. While this makes it simple to construct virtual haptic controllers, these controllers could not exhibit vibrations felt in response to the audio output.

The missing link, so to speak, has been a way of transmitting audio information back from the synthesis engine to the physical model. We previously discussed the possibility of using the JACK audio connection kit [7] to route audio data into the DIMPLE process. However, more recently we have ported DIMPLE to a PureData external [19], which has turned out to be useful for this purpose. The original intention was to help

Proceedings of ENACTIVE/07
4th International Conference on Enactive Interfaces
Grenoble, France, November 19th-22th, 2007

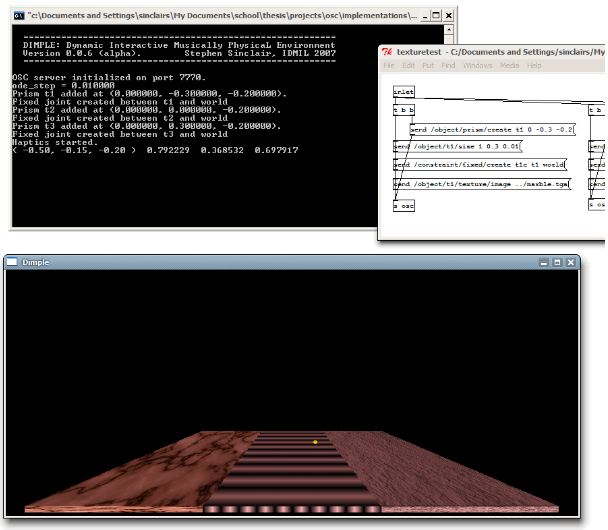

Figure 1: The PureData test patch for specifying haptic textures in DIMPLE.

in testing latency differences between using loop-back UDP/IP for OSC messages versus passing messages between threads within process memory. This has also made it possible to connect an audio patch cord in Pd to the DIMPLE object, giving access to Pd DSP stream directly. The DIMPLE-Pd's audio handler runs samples through routines from libsamplerate [8], a resampling C library, in order to downsample the audio to the haptic rate. The audio data is then transfered through a circular buffer to the haptic thread, where it can be used to modulate the force vector.

This does not minimize latency, since information must be transfered from the haptic thread to the audio thread (or process) and back. Generally, inter-modal latency (haptic events causing audio events) is more forgiving ($\tilde{2}4$ ms [1]) than purely haptic latency. Studies to determine whether or not DIMPLE's vibro-tactile latency lies below the perceptual threshold will be carried out future work.

2.2 Haptic textures

Used quite often in percussion and sound effects are interactions such as scratching, scrubbing, rolling and sliding. One way to achieve this is to perform haptic rendering of a texture, and to transmit information about the interaction stochastics back to the audio synthesis engine [22].

CHAI, which is used by DIMPLE both for haptic and visual feedback, has capabilities for rendering visual textures, but lacks support for haptically representing them. We have developed an extension which modulates the magnitude of the force vector based on a given height map. This successfully gives an impression of textures aligned with the visual bitmap because the hand can feel oscillations correlating with visual movement. Unfortunately this technique does not provide resistance to lateral movement as in a real texture. We hope to improve this in the future to support normal or height mapping, as described by Theoktisto et al. [21]. We are equally interested in exploring generative textures in 2 and 3 dimensions [10], since these can be specified on the fly with a few parameters which can easily be stored in a PureData control patch, and have advantages for continuity across surfaces.

An image of the texture interaction test can be seen in Figure 1. It shows the four messages needed to initialize each of the objects in the scene. These textures were generated synthetically using the GIMP graphics tool.

DIMPLE does not yet send information about texture interaction to the audio process. We intend to define a set of OSC messages for sending these micro-collisions as a series of impulses, or as stochastic coefficients. This information could then be used for synthesis of sonic textures, described by, for example, Barrass and Adcock [2].

2.3 Proxy object "grabbing"

CHAI works by presenting a haptic "proxy" object, a scene element that represents the location of the force-feedback controller's end effector. The algorithm used for calculating forces on the proxy is based on the *god-object* technique described by Zilles and Salisbury [24]. Since CHAI is primarily a 3-degree-of-freedom (3-DOF) system, this proxy object is presented as a sphere.

While this is adequate for touching the sides of objects and pushing on them, it leaves the user with a sense of merely "poking around" in the scene. Many gestures, musical or otherwise, depend on interaction through a tool. Examples in music include the bow, the drumstick, or the guitar pick.

A new feature in DIMPLE is to allow "grabbing" objects. When an object is grabbed, it is no longer touched by the proxy object, but instead a stiff spring is created between the object and the proxy position. This spring acts on the object in the physics thread to pull it towards the location of the proxy, but in the haptics thread opposite forces from the spring are applied to the haptic device, pulling the end effector towards the object. This has the effect of associating the object with the proxy, such that user has a sense of directly manipulating a scene object with the haptic device. The object's inertia is felt and if gravity is present, the object's weight is also distinctly perceptible. Collisions between the grabbed object and other objects are also felt in the haptic device.

The reason for using a spring between the haptics and physics threads instead of simply running the physical simulation at each haptic iteration is that it allows tight

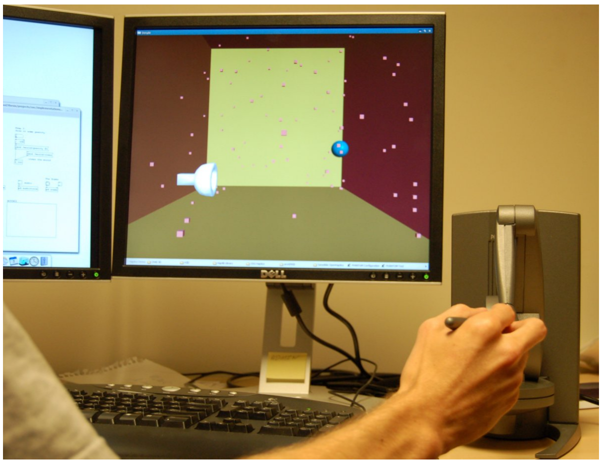

Figure 2: A demonstration of many objects bouncing around a room. The cup-shaped object is a mesh. The sphere is currently "grabbed" by the proxy, so that it is being manipulated by the haptic device. Collisions with the other objects are felt by the controller.

interaction between these two processes without sacrificing asynchrony and without requiring the demanding physics computations at haptic rates. While it may impose a lack of fidelity to the object movement, since a mass-spring system implies a certain phase delay, the response can be tuned according to the chosen physics update rate. The advantage is that the spring can be calculated at haptic rates while object movement (physics updates) can still occur at a lower speed. In same ways it is reminiscent of 6-DOF version of the *god-object* method.

2.4 Mesh objects

DIMPLE now makes use of CHAI's routines for loading 3D objects saved in the Alias Wavefront ".obj" and the 3D Studio Max ".3ds" formats. An equivalent triangular mesh is generated for ODE, as per the `dynamic_meshes` example given in the CHAI 1.61 download [6]. Mesh objects can be touched and interact in the physical environment just like prisms and spheres. Note however that mesh objects with large numbers of triangles require much more computational power than simple prisms and spheres.

Using mesh objects may open DIMPLE up to more interesting physical interactions. Oddly shaped objects may rebound in different ways than simple spheres and cubes. Grabbing mesh objects, as described above, can provide the user with a variety of "tools" that are more interesting than these simple objects. Eventually, we hope that combining mesh objects with haptic texture rendering and possibly a simple deformable vertex implementation may provide very complex and unique control surfaces.

3 Future Work

3.1 Friction models and application to bowing

One benchmark we are using to evaluate these extensions to DIMPLE is to see if it can be used to model the bowing gesture. This is because bowing has strict requirements for accurate simulation. Additionally, it is a useful baseline because it has been previously implemented several times [9, 13, 14], and there is therefore a good basis for comparison. In a companion paper at this conference [16], we suggest that all the components of torque play a part in the bowing gesture, thus the importance of implementing basic 6-DOF support in DIMPLE.

Bowing, however, also depends on a friction model designed after the interaction between a bow hair and a string. Such friction models, based on the idea of stick-slip motion, have been previously developed [9, 17] and succesfully used in virtual haptic bowing [9, 13]. Currently CHAI supports a friction model described by Zilles [23]. This does in fact render a stick-slip-enhanced Coulomb-like friction, but it is not based on the hair-string interaction as is needed for bowing. It will thus be necessary to expand CHAI to support various other friction models so that physical models such as Serafin's bowing synthesis can be explored.

Additionally, it is not clear how the friction haptic model might interact with the 6-DOF physics engine, since they operate independently in DIMPLE. At this time, friction simulation is lost when a "grabbed" object acts as the proxy.

4 Conclusion

We have described several enhancements to DIMPLE that allow more intricate interaction techniques. It is hoped that these additions will allow the creation of interesting haptic virtual instruments based on known musical gestures such as bowing.

While the slower rigid body model in DIMPLE has advantages in terms of computational demand and allows the use of common computing hardware to maintain a real-time multi-modal simulation, it unfortunately creates artificial limits on the fidelity of interaction at the texture and friction level. It is hoped that by implementing textures and friction models in the haptic thread, and by creating a pipeline between the audio synthesis engine and the haptic simulation, this fidelity can be restored.

Since the haptics thread implements a 3-DOF collision algorithm, the dependance on the physics model for 6-DOF haptic computation may turn out to be a limiting factor in allowing this level of interaction for 6-DOF haptics. With triangle meshes in particular, especially between textured surfaces, 6-DOF haptic rendering requires significant optimization and is still an active area of research. (See, for example, Otaduy et al. [15].) However, for simple models it may very well be possible to turn up the crank and run the physics simulation at haptic rates.

5 Acknowledgements

This work was sponsored in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canadian Foundation for Innovation (CFI), and the Enactive Network European Project.

References

[1] B. D. Adelstein, D. R. Begault, M. R. Anderson, and E. M. Wenzel:. Sensitivity to haptic-audio asynchrony. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, pages 73–76, Vancouver, BC, November 2003. ACM.

[2] S. Barrass and M. Adcock. Interactive granular synthesis of haptic contact sounds. In *Proceedings of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, Helsinki University of Technology, Espoo, Finland, June 2002.

[3] C. Cadoz, A. Luciani, and J. L. Florens. CORDIS-ANIMA: a modeling and simulation system for sound and image synthesis—the general formalism. *Computer Music Journal*, 17(1):19–29, 1993. MIT Press.

[4] C. Chafe. Tactile audio feedback. In *Proceedings of the International Computer Music Conference*, pages 76–79. ICMA, 1993.

[5] F. Conti, , F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. M. L. Sentis, E. Vileshin, J. Warren, O. Khatib, and K. Salisbury. The CHAI Libraries. In *Eurohaptics '03*, pages 496–500, Dublin, Ireland, June 2003.

[6] F. Conti, D. Morris, F. Barbagli, and C. Sewell. CHAI 3D. Available: `http://www.chai3d.org/`, November 2006.

[7] Davis, P. et al. JACK Audio Connection Kit (software). Available: `http://jackaudio.org`.

[8] E. de Castro. libsamplerate (software), August 2007. `http://www.mega-nerd.com/SRC/`.

[9] J.-L. Florens. Expressive bowing on a virtual string instrument. In A. Camurri and G. Volpe, editors, *Lecture Notes in Computer Science: Gesture-Based Communication in Human-Computer Interaction, 5th International Gesture Workshop*, volume 2915, pages 487–496. Springer Berlin / Heidelberg, Genova, Italy, April 2003.

[10] J. P. Fritz and K. E. Barner. Stochastic models for haptic texture. In M. R. Stein, editor, *Proceedings of SPIE*, volume 2901, pages 34–44. SPIE, 1996.

[11] B. Gillespie. Haptics. In P. R. Cook, editor, *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*, chapter 13, pages 229–246. MIT Press, Cambridge, MA, March 1999.

[12] M. Minsky, O.-Y. Ming, O. Steele, F. P. Brooks Jr., and M. Behensky. Feeling and seeing: issues in force display. In *SI3D '90: Proceedings of the Symposium on Interactive 3D Graphics*, pages 235–241, New York, NY, USA, 1990. ACM Press.

[13] C. Nichols. The vBow: development of a virtual violin bow haptic human-computer interface. In *Proceedings of the Conference on New Interfaces for Musical Expression*, pages 29–32, Dublin, Ireland, 2002.

[14] S. M. O'Modhrain. *Playing by Feel: Incorporating Haptic Feedback into Computer-Based musical Instruments*. PhD thesis, Stanford University, November 2000.

[15] M. A. Otaduy, N. Jain, A. Sud, and M. C. Lin. Haptic display of interaction between textured models. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 133, New York, NY, USA, 2005. ACM Press.

[16] E. Schoonderwaldt, S. Sinclair, and M. M. Wanderley. Why do we need 5-dof force feedback? the case of violin bowing. In *Proceedings of Enactive '07*, 2007. Submitted for approval.

[17] S. Serafin, C. Vergez, and X. Rodet. Friction and application to real-time physical modeling of a violin. In *Proceedings of the International Computer Music Conference*, pages 375–377, Beijing, China, 1999. ICMA.

[18] S. Sinclair and M. M. Wanderley. Defining a control standard for easily integrating haptic virtual environments with existing audio/visual systems. In *Proceedings of the Conference on New Interfaces for Musical Expression*, pages 209–212, New York, NY, June 2007.

[19] S. Sinclair and M. M. Wanderley. Using PureData to control a haptically-enabled virtual environment. In *PureData Convention '07*, Montreal, Canada, August 2007. Pending publication.

[20] R. Smith. Open Dynamics Engine (software). Available: `http://www.ode.org`, November 2006.

[21] V. Theoktisto, M. Fairén, I. Navazo, and E. Monclús. Rendering detailed haptic textures. In F. Ganovelli and C. Mendoza, editors, *Workshop On Virtual Reality Interaction and Physical Simulation*. Eurographics Association, 2005.

[22] K. van den Doel, P. G. Kry, and D. K. Pai. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on computer graphics and interactive techniques*, pages 537–544, 2001.

[23] C. Zilles. Haptic rendering with the toolhandle haptic interface. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[24] C. Zilles and J. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 3, page 3146, 1995.