Virtual Intimacy : Phya as an Instrument

Dylan Menzies
Dept. Computer Science and Engineering
De Montfort University
Leicester, UK
rdmg dmu.ac.uk

ABSTRACT

Phya is an open source C++ library originally designed for adding physically modeled contact sounds into computer game environments equipped with physics engines. We review some aspects of this system, and also consider it from the purely aesthetic perspective of musical expression.

Keywords

NIME, musical expression, virtual reality, physical modeling, audio synthesis

1. INTRODUCTION

The use of impact sounds coupled to a modeled environment was introduced in [5]. Refinements of impact sound models have since been made [1] . The first working models for sustained contact sounds integrated with a physical environment was made in [13], greatly expanding the overall realism of the simulation by relating audio and visual elements continuously. Frictional models have been created for musical instruments, and have also been applied to surfaces in simulated environments [2]. Further models have have been proposed for other environmental sounds including fluids [12]. In [7] a framework was presented for a physical audio system, designed to operate closely with a physics engine providing rigid body dynamics. The emphasis was on using robust techniques that could be scaled up easily, and accommodate an environment that was rapidly changing. This work developed into the Phya physical audio library discussed here.

The principle goal for Phya has been to generate sounds that arise from dynamical interactions, that are either that are clearly visually apparent, or directly affected by user control. This is because when audio can be closely causally correlated to other percepts, the overall perceptual effect and sense of immersion is magnified considerably. A wide selection of sounds fall into this category, including collisions between discrete solid and deformable objects. The complex dynamics of these objects is captured well by the many physics engines that have been developed. The audio generated is a modulation of the audio rate dynamics of excitation and resonance, by the relatively slow large scale dynamics of objects. Simple audio synthesis processes can lead to convincing results, but only if coupled carefully with the large scale dynamics.

While physical modeling provides a powerful way to generate strong percepts, a balance must be struck on the level of detail, so that the output is not overly constrained. In practice this leads to the development of semi-physical-perceptual models that provide some freedom for the sound designer to more easily mould a virtual world.

It was apparent from early on, that Phya offers an inherently musical experience, even from the limited control environment of a desktop. The richness of dynamic behavior and multi-modal feedback are characteristic of musical performance. A later section explores this further. Use of coupled musical-visual performances has become common, however performances within a physical audio-visual world are still apparently scarce, as are physical audio-visual worlds in computer games. This state of affairs has prompted this article.

NIME08, Genova, Italy

2. TECHNOLOGICAL REVIEW

Below we briefly review the components of Phya, and the overall structure used to accommodate them.

2.1 Impacts

2.1.1 Simple spring

The simplest impacts consist of a single excitation pulse, which then drives the resonant properties of the colliding objects. The spectral brightness of the pulse depends on the combined hardness of the two surfaces. Using a spring model, the combined spring constant, which determines the duration and so spectral profile of a hit, is $k = (k_1^{-1}+k_2^{-1})^{-1}$ where k_1 and k_2 are the spring constants of the individual surfaces. A model which just takes k to be the lesser value is also adequate. The duration is $\pi\sqrt{m/k}$ where m is the effective mass $(m_1^{-1} + m_2^{-1})^{-1}$. The effective mass can be approximated by the lesser mass. If one object is fixed like a wall, the effective mass is the free object's mass.

The impact displacement amplitude in this model is, $A = v\sqrt{m/k}$ where v is the relative normal contact speed. To give the sound designer more freedom over the relation between collision parameters and the impact amplitude, a linear breakpoint scheme is used with an upper limit also providing a primary stage of audio level limiting. Note that the masses used for impact generation do not have to be in exact proportion to the dynamics engine masses.

Audio sensitivity to surface hardness and object mass, helps to paint a clearer picture of the environment. From a musical perspective it adds variation to the sound that can be generated, in an intuitive way.

2.1.2 Stiffness

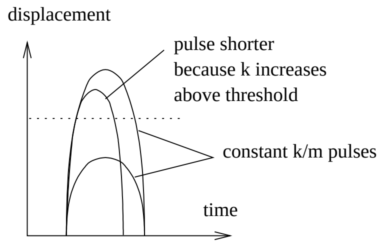

Figure 1: Displacements from three impacts, one of which is stiff.

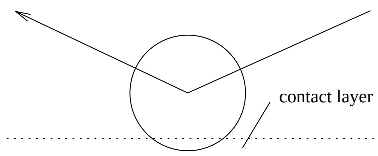

Figure 2: A grazing impact.

Impact stiffness is important for providing cues to the listener about impact dynamics, because it causes spectral changes in the sound depending on impact strength, whereas impact strength judged from the amplitude level of an impact received by a listener is ambiguous because of the attenuating effect of distance. Stiffness can be modeled by making the spring constant increase with impact displacement. This causes an overall decrease in impact duration for an increase in impact amplitude, and makes it spectrally brighter, illustrated in Figure 1. The variation in stiffness with impulse is a property of the surface and can be modeled reasonably well with a simple breakpoint scheme, that can be tuned by the sound designer directly.

Increasing brightness with note loudness is an important attribute of many musical instruments, acoustic and electronic, and is rooted in our everyday physical experience. It might even be called a universal element of expression. Phya incorporates this behavior naturally.

2.1.3 *Multiple hits and grazing*

Sometimes several hits can occur in rapid succession. A given physics engine would be capable of generating this impact information down to a certain time scale. The effect can be simulated by generating secondary impulses according to a simple poisson-like stochastic process, so that for a larger impact the chance of secondary impacts increases. Also common are grazing hits, in which an impact is associated with a short period of rolling and sliding. This is because the surfaces are uneven, and the main impulse causing the rebound occurs during a period of less repulsive contact. Such fine dynamics cannot be captured by a typical physics engine. However, good results can be achieved by combining an audio impulse generation with a continuous contact generation, according to the speed of collision and angle of incidence, see Figure 2. The component of velocity parallel to the surface is used for the surface contact speed.

2.2 Continuous contacts

2.2.1 *Basic model*

Continuous contact generation is a more complex process. The first method introduced, [13], was to mimic a needle following the groove on a record. This corresponds to a contact point on one surface sliding over another surface, and is implemented by reading or generating a surface profile at the contact point to generate an audio excitation.

Rolling is similar to sliding, except there is no relative movement at the contact point, resulting in a spectrally less bright version of the sliding excitation. This can be modeled by appending a lowpass filter that can be varied according to the slip speed at the contact, creating a strong cue for the dynamics there. See Figure 3. A second order filter is useful to shape the spectrum better. The contact excitation is also amplified by the normal force, in the same way impacts are modified by collision energy. More subtle are modifications to spectral brightness according to the m/k ratio that determines the brightness of an impact. Low m/k corresponds to a light needle reading the surface at full brightness. Heavier objects result in slower response, which can modeled again by controlling the lowpass filter. Although simple, this efficient model is effective because it takes in the full dynamic information of the contact and uses it to shape the audio which we then correlate with the visual portrayal of the dynamics. It is also easily customized to fit the sound designers requirements. When flat surfaces are in contact over a wide area this can be treated as several spaced out contact points, which can often be supplied directly by the dynamics-collision system.

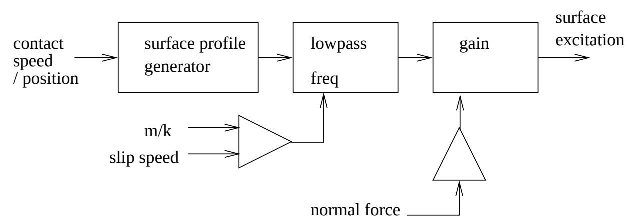

Figure 3: Surface excitation from rolling and sliding.

2.2.2 *Contact jumps*

Even for a surface that is completely solid and smooth, the excitations do not necessarily correspond very well with the surface profile. A contact may jump creating a small micro-impact, due to the blunt nature of the contact surfaces, see Figure 4. The sound resulting from this is significant and cannot be produced by reading the surface profile directly. Again, the detailed modeling of the surface interactions is beyond the capabilities available from dynamics and collisions engines, which are not designed for this level of detail. Good results can instead be achieved by adding the jumps, pre-processed, into the profile, Figure 5. Downsampling a jump results in a bump, unless it is sampled with sufficient initial resolution, which may be impractical. A useful variation is therefore to downsample jumps to jumps, by not interpolating. This retains the 'jumpiness' and avoids the record-slowing-down effect.

2.2.3 *Programmatic and stochastic surfaces*

Stored profiles can be mapped over surface areas to create varying surface conditions. This can be acceptable for sparse jump-like surfaces that can be encoded at reduced sample rates, but in general the memory requirements can be unreasonable. An alternative is to describe surfaces programmatically, either in a deterministic or fully stochastic way. The advantage of a largely deterministic process is that repetitions of a surface correlate closely, for instance when something is rolling back and forth, providing consistency cues to the dynamic behavior even without visuals. Indexable random number generators provide a way to deterministically generate random surfaces. Others include

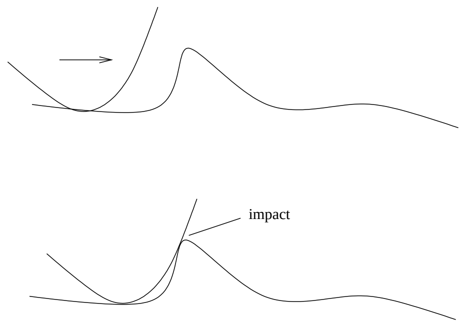

Figure 4: Micro-impact occuring due to contact geometry

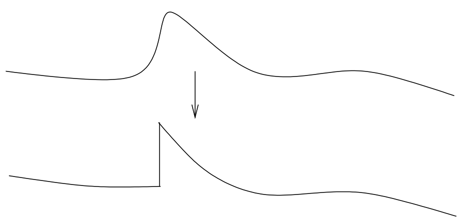

Figure 5: Preprocessing a surface profile to include jumps.

repeating functions to generate pattern based surfaces such as grids.

A useful range of surfaces can be generated by stochastically generating pulses of different widths, with control over the statistical parameters. A change of contact speed is then achieved by simply varying the parameters.

Secondary excitations can also be generated stochastically, for instance to simulate the disturbance of gravel on a surface, in a similar manner to the physically informed footsteps in [3], Figure 6. In this scheme the collision pa-

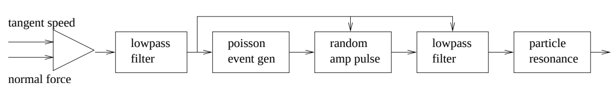

Figure 6: Modeling loose surface particle sound.

rameters are used to determine the activity rate of a poisson like process which then generates impulses mimicking the collisions of gravel particles. A low frequency lowpass filter is used to simulate the duration of the particle spray following an impact. The impulses have randomly selected amplitudes and are shaped or filtered to reflect increased particle collision brightness with increased contact force and speed, before exciting a particle resonance. This model simplifies the fact that at high system collision energies there will still be particle collisions occurring at low energy. It also assumes all particles have the same resonance. The model does however have sufficient dynamic temporal and spectral behavior to be interesting. Three levels of dynamics can be distinguished here, the gross object dynamics, the simulated gravel dynamics, and audio resonance. The detail that can be encoded in surface excitations is critical from the musical point of view. It provides the foundation from which the full sounds evolves.

2.2.4 Friction

Friction stick and slip processes are important in string instruments. In virtual environments they are much less common source of sound than the interactions considered so far. A good example is door creaking, which is visually linked to the movement of the door. Stick and slip for discrete solid objects is simulated well by the generation of pulses at regular linear or angular intervals. The amplitude and spectral profile of the pulses modifying as the contact force and speed changes. As contact force increases, normally the interval between each pulse increases, due to the increased static friction limit, with more or less constant lateral spring constant.

2.2.5 Buzzing

Common phenomena are buzzing and rattling at a contact, caused by objects in light contact that have been set vibrating. Like impact stiffness, it provides a distant independent cue of dynamic state, which in this case is the amplitude of vibration. Objects that are at first very quiet can become loud when they begin to buzz, due to the nonlinear transfer of low frequency energy up to higher frequencies that are radiated better. Precise modeling of this with a dynamics-collision engine would be infeasible. However, the process can be modeled well by clipping the signal from the main vibrating object, as shown in Figure 7, and feeding it to the resonant objects that are buzzing against each other. The process could be made more elaborate by calculating the mutual excitation due to two surfaces moving against each other.

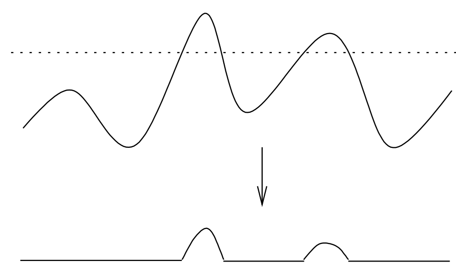

Figure 7: Clipping of resonator output to provide buzz excitation.

2.3 Resonators

2.3.1 Modal resonators, calibration, location dependence

There are many types of resonator structure that have been used to simulate sounding objects. For virtual environments we require a minimal set of resonators that can be easily adapted to a wide variety of sounds, and can be efficiently run in numbers. The earliest forms of resonator used for this purpose were modal resonators [5, 13] which consist of parallel banks of second order resonant filters, each with individual coupling constants and damping. These are particularly suited to objects with mainly sharp resonances such as solid objects made from glass, stone and metal. It is possible to identify spectral peaks in the recording of a such an object, and also the damping by tracking how quickly each peak decays, [11]. A command line tool is included with Phya for automating this process. The resultant data is many times smaller than even a single collision sample.

Refinements to this process included sampling over a range of impact points, and using spatial sound reconstruction. The associated complexities were not considered a priority in Phya. Hitting an object in different places produces different sounds, but just hitting an object in the same place repeatedly produces different sounds each time, due to the changing state of the resonant filters. It is part of the attraction of physical modeling that such subtleties are manifested. If needed, an collision object can be broken up into several different collision objects, and different Phya sound objects associated with these.

2.3.2 Diffuse resonance

For a large enough object of a given material the modes become very numerous and merge into a diffuse continuum. This coincides with the emergence of time domain structure at scales of interest to us, so that for instance a large plate of metal can be used to create echos and reverberation. For less dense, more damped material such as wood, pronounced diffuse resonance occurs at modest sizes, for instance in chairs and doors. Such objects are very common in virtual environments and yet a modal resonator is not efficiently able to model diffuse resonance, or be matched to a recording. Waveguide methods have been employed to model diffuse resonance either using abstract networks, including banded waveguides [4], feedback delay networks [9] or more explicit structures such as waveguide meshes [14, 15]. An alternative approach introduced in [6], is to mimic a diffuse resonator by dividing the excitation into frequency bands, and feeding the power in each into a multi-band noise generator, via a filter that generates the time decay for each band, see figure 8. This *perceptual resonator* provides a diffuse response that responds to the input spectrum. When combined with modal modeling for lower frequencies it can efficiently simulate wood resonance, and can be easily manipulated by the sound designer. A similar approach had been used in [10] to simulate the diffuse resonance of sound boards to hammer strikes, however the difference here is that the resonator follows the spectral profile of a general input.

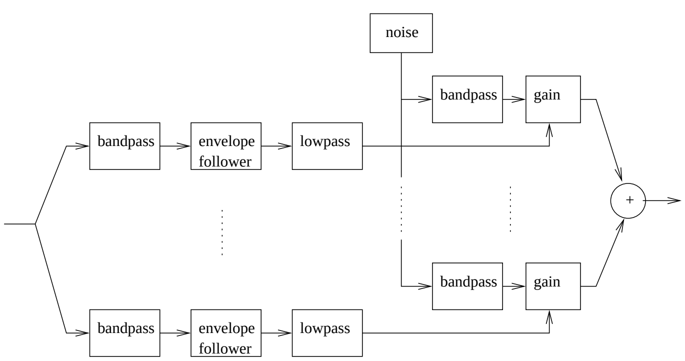

Figure 8: Outline of a perceptual resonator.

2.3.3 Surface damping

A common feature of resonant objects is that their damping factors are increased by contact with other objects. For instance a cup placed on a table sounds less resonant when struck. This behavior has a strong visual-dynamic coupling, and provides information about the surfaces. It can be simulated by accumulating a damping factor for each resonator as a sum of damping factors associated with the surfaces that are in contact.

2.3.4 Nonlinear resonance

Many objects enter non-linear regimes when vibrating strongly, sometimes causing a progressive shift of energy to higher frequencies. For a modal system this can be modeled by exciting higher modes by lower modes via nonlinear couplings. In waveguide systems the non-linearities can be built into the network.

2.3.5 Deformable objects

There is a common class of objects that are not completely rigid, but still resonate clearly, for example a thin sheet of metal. Such objects have variable resonance characteristics depending on their shape. While explicit modeling of the resonance parameters according to shape is prohibitive, an excellent qualitative effect that correlates well with visual dynamics is to vary the resonator parameters about a calibrated set, according variations of shape from the nominal. This can be quantified in a physical model of a deformable model by using stress parameters or linear expansion factors. The large scale oscillation of such a body modulates the audio frequencies providing an excellent example of audiovisual dynamic coupling.

2.4 Phya overall structure and engine

Phya is built in the C++ language, and is based around a core set of general object types, that can specialized and extended. Sounding objects are represented by a containing object called a *Body*, which refers to an associated *Surface* and *Resonator* object, see Figure 9. Specializations of these include *SegmentSurface* for recorded surface profiles, *RandSurface* for deterministically generated stochastic surfaces, *GridSurface* for patterns. The resonator subtypes are *ModalResonator* and *PerceptualResonator*. Bodies can share the same surface and resonator if required in order to handle groups of objects more efficiently. Collisions states are represented using *Impact* and *Contact* objects that are dynamically created and released as collisions occur between physical objects. These objects take care of updating the state of any associated surface interactions.

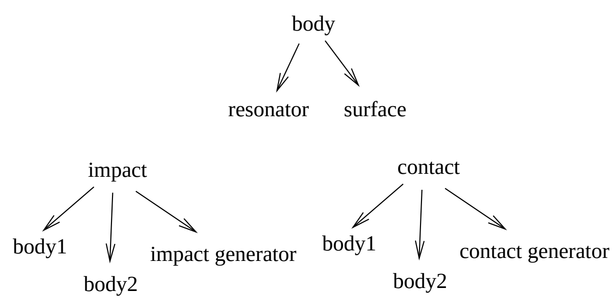

Figure 9: Main objects in Phya.

2.4.1 System view

The top level system view is shown in Figure 10. The collision system in the environment simulator must generate trigger updates in Phya's collision update section, for example using a callback system. This in turn reads dynamic information from the dynamics engine and updates parameters that are used by the Phya audio thread to generate audio samples. The most awkward part of the process is finding a way for Phya to keep track of continuous contacts.

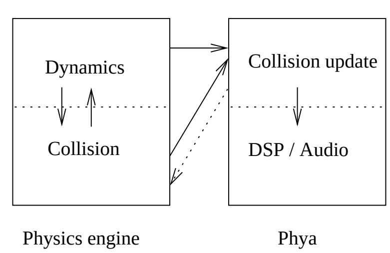

Figure 10: Phya system overview.

2.4.2 Tracking contacts

Most collision engines do not use *persistent contacts*, meaning they forget information about contacts from one collision frame to another. On the other hand Phya wishes to remember contacts because it has audio processes that generate excitations continuously during a contact. The problem can be attacked either by modifying the collision engine, which is hard or not possible, or searching contact lists. In the simplest case, the physics engine provides a list of non-persistent physical contacts at each collision step, and no other information. For each physical contact, the associated Phya bodies can be found and compared with a list of current Phya contact pairs. If no pair matches a new Phya contact is formed. If a pair is found, it is associated with the current physical contact. For any pairs left unmatched, the associated Phya contact is released. See Figure 11. This works on the, mostly true, assumption that if a physical contact exists between two bodies in two successive frames then that is a continuous contact evolving. If two bodies are in contact in more than one place then some confusion can occur, but this is offset by the fact that the sound is more complex. Engines that keep persistent contacts are easier to handle. The ability to generate callbacks when contacts are created and destroyed helps even more.

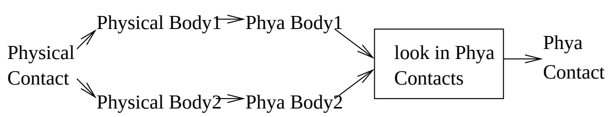

Figure 11: Find a Phya contact from a Physical contact.

2.4.3 Smooth surfaces

Another problem of continuous contacts arises from the collision detection of curved surfaces. For example the collision of a cylinder can be detected using a dedicated algorithm, or a more general one applied to a collision net that approximates a cylinder. From a visual dynamic point of view the general approach may appear satisfactory. However, the dynamic information produced may lead to audio that is clearly consistent with an object with corners and not smooth. A way to improve this situation is to smooth the dynamic information when it is intended that the surface is smooth, using linear filters. This requires Phya to check the tags on the physical objects associated with a new contact to see if smoothing is intended.

2.4.4 Limiters

The unpredictable nature of physical environmental sound requires automated level control, both to ensure it is sufficiently audible and also not so loud to dominate other audio sources or to clip the audio range. This has already been partly addressed at the stage of excitation generation, however because of the unpredictability of the whole system, it is also necessary to apply limiters to the final mix. This is best achieved with a short look-ahead brick wall limiter, that can guarantee a limit, while also reducing annoying artifacts that would be caused without any look-ahead. Too much look-ahead would compromise interactivity, however the duration of a single audio system processing vector, which is typically 128 samples, is found to be sufficient.

3. A VIRTUAL MUSICAL INSTRUMENT

While Phya was designed for general purpose virtual worlds, the variety and detail of sonic interactions on offer lend themselves to the creation of musical virtual instruments. From a more abstract view, the layered, multi-scale dynamics within Phya capture the layered dynamics present in real acoustic instruments. It is sometimes claimed that this structure is particularly relevant to musical performance, [8]. Electronic performance systems often fail to embody the full range of dynamic scales, even within physically modeled instruments, which sometimes lack physical control interfaces with appropriate embedded dynamics.

Although grounded in physical behavior, and therefore naturally appealing to human psychology, the intimate interactions can be tailored to more unusual simulations that would be difficult or impossible in the real world. For instance very deep resonances can be easily created that would require very heavy objects, and unusual resonances can be created. Likewise, the parameters of surfaces can be composed to ensure the desired musical effect. The physical behavior of objects can be matched to any desired scale, of distance, time or gravity. Because the graphical world is virtual it too can be composed artistically with more freedom than the real world.

The graphical output not only provides additional feedback to the performer, but adds the kind of intimate visual association, present in traditional musical performance, but lacking in much live electronic music, especially that focused around keyboard and mouse control. Phya provides the audience with an alternative to the performer as a visual focus.

The mouse interface is readily extended to a more haptically and visually appropriate controller using a device such as a Nintendo Wii remote. This has the effect of making the control path correspond directly to the object path, improving the sense of immersion for the performer. In a CAVE like environment the performer can maneuver within a spatial audio environment, although without an audience. In a full headset virtual reality environment, the performer can interact directly with objects through virtual limbs, with virtual co-performers and virtual audience.

While Phya has not been used yet to produce an extended musical work, we discuss musical aspects of some demonstrations. Figures 12 and 13, show simple examples of sonic toys constructed with Phya. In the first nested spheres form a kind of virtual rattle, with the lowest resonance associated with the biggest sphere. The user interacts by dragging the middle sphere around by invisible elastic. The second shows a deformable teapot with a range of resonances. The deformation parameters are used to modify the resonant frequencies on the fly. The effect is at once familiar and surreal. Further examples demonstrate the stacking of many different resonant blocks. Configuring groups of blocks becomes a musical, zen-like process.

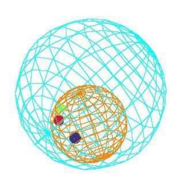

Figure 12: Nested sonic spheres.

4. COPING WITH NETWORK LATENCY

There has been considerable interest in collaborative interactive musical performance over networks. One aspect of such systems is the delay or latency required to transmit information around the network, which can be musically significant for long distance collaborations. In the case of

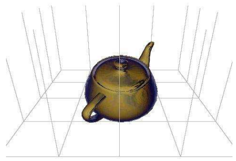

Figure 13: Deformable sonic teapot.

performance with acoustic instruments, it is impossible to make each side hear the same total performance while also playing their instruments normally. Virtual instruments of the kind described here offer another possibility, due to the fact that the dynamics of the virtual world is strictly separated from the control in the outer world. Figure 14 shows a collaboration between two performers across a network. Adding local delays to match the network latency keeps the

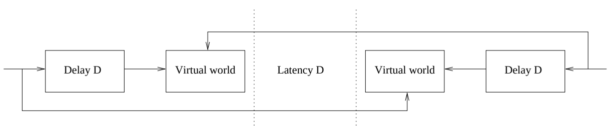

Figure 14: Two performers with local virtual worlds.

two virtual worlds synchronized. In each world the audio and graphical elements are of course synchronized. Performance gestures are delayed, but this is not such a severe handicap because the visual feedback remains synchronized, and is a price worth paying to maintain overall synchronization over the network. If control is by force rather than position, the gesture delay is even less intrusive. To eliminate drift between the virtual worlds, and handle many performers efficiently, a central virtual world can be used, shown in Figure 15 This adds return latency delays.

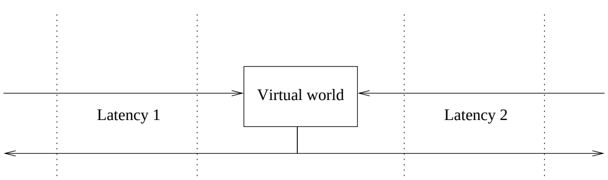

Figure 15: Many performers with a central virtual world.

5. BACK TO REALITY

The aesthetics of Phya partly inspired a tangible musical performance piece, that we mention briefly because it provides an interesting example of how the boundary between virtual and real can become blurred. *Ceramic Bowl*[1] centers around a bowl with 4 contact microphones attached around the base, where there is a hole. Objects are launched manually into the bowl where they roll, slide and collide in orbit until they exit. The captured sound is computer processed under realtime control and diffused onto an 8 speaker rig. The microphone arrangement allows the spatial sound events to be magnified over a large listening area.

[1]First performed at the Electroacoustic Music Studies conference, Leicester, 14 June 2007. Broadcast on BBC Radio 3 Hear and Now, 25 August 2007

6. CONCLUSION

The original goal was to create a system that can capture the sonic nuance and variety of collisions, and that was easy to configure and use within a virtual reality context. This required the consideration of a variety of inter-dependent factors. The result is a system that is not only useful from the point of view of virtual reality, but has natural aesthetic interest and application in musical performance. The integrated graphical output is part of a fused perceptual aesthetic. Phya is now an open source project. [2].

7. REFERENCES

[1] F. Avanzini, M. Rath, and D. Rocchesso. Physically-based audio rendering of contact. In *Proc. IEEE Int. Conf. on Multimedia and Expo, (ICME2002), Lausanne*, volume 2, pages 445–448, 2002.

[2] F. Avanzini, S. Serafin, and D. Rocchesso. Interactive simulation of rigid body interaction with friction-induced sound generation. *IEEE Tr. Speech and Audio Processing*, 13(5.2):1073–1081, 2005.

[3] P. Cook. Physically informed sonic modeling (phism): Synthesis of percussive sounds. *Computer Music Journal*, 21:3, 1997.

[4] G. Essl, S. Serafin, P. Cook, and J. Smith. Theory of banded waveguides. *Computer Music Journal*, spring 2004.

[5] J. K. Hahn, H. Fouad, L. Gritz, and J. W. Lee. Integrating sounds and motions in virtual environments. In *Sound for Animation and Virtual Reality, SIGGRAPH 95*, 1995.

[6] D. Menzies. Perceptual resonators for interactive worlds. In *Proceedings AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, 2002.

[7] D. Menzies. Scene management for modelled audio objects in interactive worlds. In *International Conference on Auditory Display*, 2002.

[8] D. Menzies. Composing instrument control dynamics. *Organized Sound*, 7(3), April 2003.

[9] D. Rochesso and J. O. Smith. Circulant and elliptic feedback delay networks for artificial reverberation. *IEEE trans. Speech and Audio*, 5(1):1997, 1997.

[10] J. O. Smith and S. A. Van Duyne. Developments for the commuted piano. In *Proceedings of the International Computer Music Conference, Banff, Canada*, 1995.

[11] K. van den Doel. *Sound Synthesis for Virtual Reality and Computer Games*. PhD thesis, University of British Columbia, 1998.

[12] K. van den Doel. Physically-based models for liquid sounds. *ACM Transactions on Applied Perception*, 2:534–546, 2005.

[13] K. van den Doel, P. G. Kry, and D. K. Pai. Foleyautomatic: Physically-based sound effects for interactive simulation and animation. In *Computer Graphics (ACM SIGGRAPH 01 Conference Proceedings)*, 2001.

[14] S. A. Van Duyne and J. O. Smith. Physical modeling with the 2-d digital waveguide mesh. In *Proc. Int. Computer Music Conf., Tokyo*, 1993.

[15] S. A. Van Duyne and J. O. Smith. The 3d tetrahedral digital waveguide mesh with musical applications. In *Proceedings International Computer Music Conference*, 2001.

[2]Details at www.zenprobe.com/phya