PHYSICAL AUDIO FOR VIRTUAL ENVIRONMENTS, PHYA IN REVIEW

Dylan Menzies

Dept. Computer Science and Engineering
De Montfort University
Leicester UK
dylan@dmu.ac.uk

ABSTRACT

A review is presented of a library that has emerged out of the development of physical audio capability within a physical computer game environment. Technical aspects are covered with emphasis on practical requirements, as well as broader issues concerning the uptake of audio modeling within industry. Some future directions are considered.

1. INTRODUCTION

The use of impact sounds coupled to a modeled environment was introduced in [1]. Refinements of impact sound models have since been made [2] . The first working models for sustained contact sounds integrated with a physical environment was made in [3], 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 [4]. Further models have have been proposed for other environmental sounds including fluids [5]. In [6] 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 expanded easily, and acomodate an environment that was rapidly changing. This work developed into the Phya physical audio library discussed here. An overview is presented of the processes on which it is based, its structure, and recent developments. It is important to consider the library in tandem with the physical simulation with which it must operate. The dynamics of both the physics engine and Phya combine to provide the final auditory result. Refinements of the audio modeling process needs to be balanced by an understanding of the features emerging from the composite system that provide the best audio cues to the listener. Coupled to this is often a need to be able to creatively control the characteristics of the system without imposing strict physical constraints, and retaining the important behaviour from a perceptual point of view. When the dynamics of some objects, conveyed through graphics, correlates closely with the sound generated, a powerful synergistic perceptual effect is produced. However, where the visual link to a sound becomes blurred or less obvious, then explicit audio modeling techniques become of less value, and more programmatic techniques may be more appropriate.

2. TECHNOLOGICAL REVIEW

The scope of Phya has so far been limited to sound resulting from the interactions between discrete solid and deformable objects, and mirrors the development of dynamics-collision engines. Interactions between discrete objects account for large proportion of everyday environmental sounds, and merit accurate modeling because the sound generated greatly enhances our knowledge of dynamics gained from visuals alone. Below we review the models that are built into Phya, and consider the overall structure used to accomodate them. Possible extensions to cloth and water models are also discussed.

2.1. Impacts

2.1.1. *Spring model*

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 can also be calculated easily in this model, $A = v\sqrt{m/k}$ where v is the relative normal contact speed, however in practice the sound designer requires more freedom over the relation between collision parameters and the impact amplitude. This can be specified by a linear breakpoint scheme 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 directly match the dynamics engine masses.

2.1.2. *Stiffness*

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 pulse shape can be controlled by the direct synthesis of the pulse including its duration and amplitude. Similar results are obtained by controlling a 1st order lowpass filter fed by a fixed sharp pulse. Precise modeling is not required, in fact this is an obstacle to tuning the behaviour according to the sound designer's

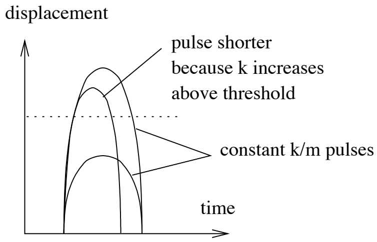

Figure 1: Displacements from three impacts, one of which is stiff.

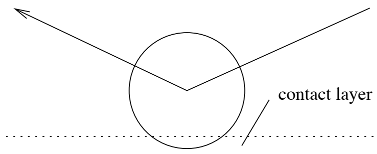

Figure 2: A grazing impact.

requirements. For instance they might require stiffness to become prominent for a certain impact strength, and independently control the nominal hardness.

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, [3], 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 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 here 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,

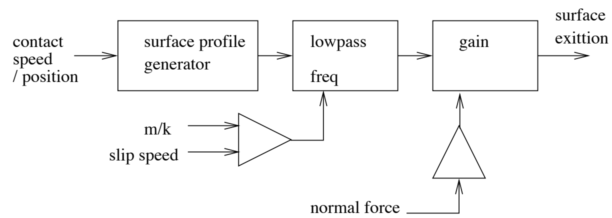

Figure 3: Surface excitation from rolling and sliding.

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.

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.

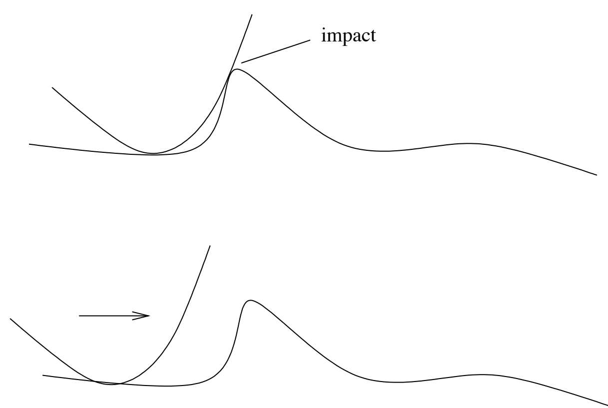

Figure 4: Micro-impact occuring due to contact geometry

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, even in todays

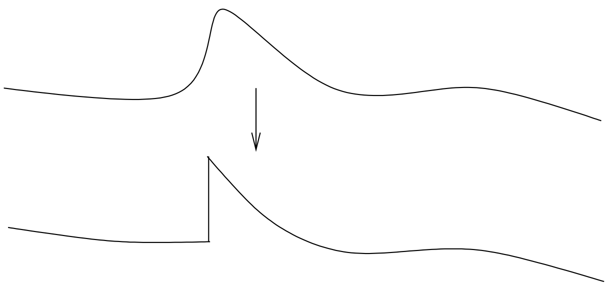

Figure 5: Preprocessing a profile to include jumps.

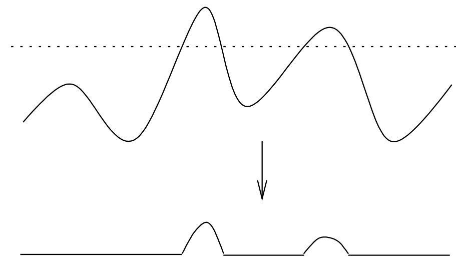

Figure 6: Clipping of resonator output to provide buzz excitation.

games consoles. An alternative is to describe surfaces programmatically, either in a deterministic or 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 a strong cue to the dynamic behaviour even without visuals. Indexable random number generators provide a way to deterministically generate surfaces. Others include repeating functions to generate pattern based surfaces such as grids. Stochastic processes can be used to model the surface directly, for instance white noise is an effective surface. 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 [7]. In this scheme the primary excitations are filtered to determine the activity rate of a poisson like process which then generates impulses mimicking the collisions of gravel particles.

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 strongly coupled to the visual portrayal of the dynamics. Stick and slip for discrete solid objects is simulated well by the generation of pulses at regular linear or angular intervals determined by the normal contact force, with the amplitude and spectral profile of the pulses modifying as the contact force and speed changes. As contact force increases, so the interval between each pulse increases, due to static friction.

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 6, 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.

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 [1, 3] 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, [8]. In other words the objects resonant behaviour can be physically sampled. 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 important 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 [9], feedback delay networks [10] or more explicit structures such as waveguide meshes [11, 12]. An alternative approach introduced in [13], 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 7. 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 [14] to simulate the diffuse resonance of sound boards to hammer strikes, however the difference is that our resonator follows the spectral profile of a general input.

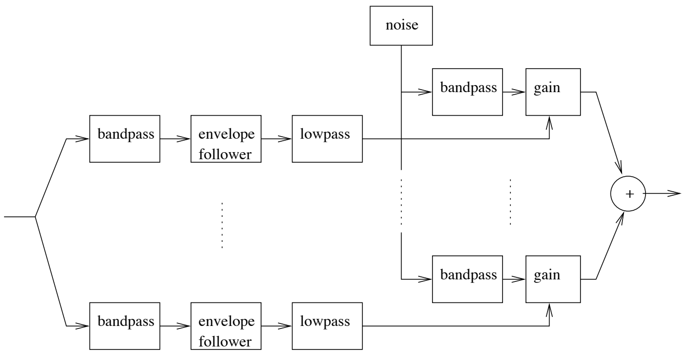

Figure 7: Outline of a perceptual resonator.

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 so is well suited for simulation, which can be achieved 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, usually causing a spectral shift to higher frequencies, and generating a more complex sound overall. This is related to the buzzing discussed earlier in that different linear resonators can be coupled non-linearly via their surfaces. These processes might actually be present within a given object. A first approximation is to distort any resonator with a nonlinear function. This can be made more realistic by feeding back some of the distorted signal into the resonator, see Figure 8.

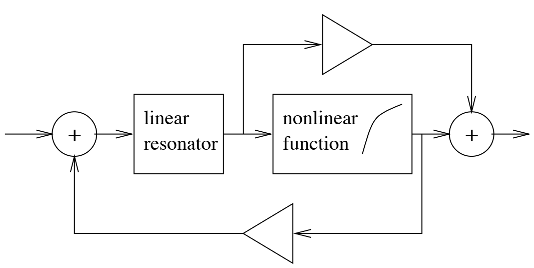

Figure 8: Nonlinear resonator

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 modelling 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.

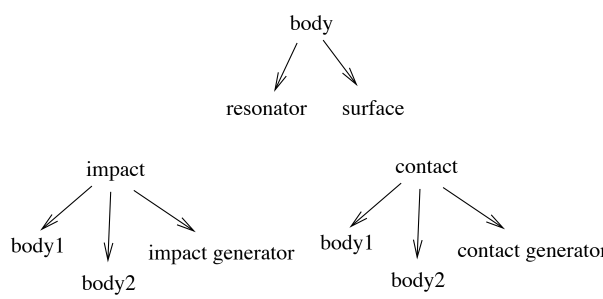

Figure 9: Main objects in Phya.

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.

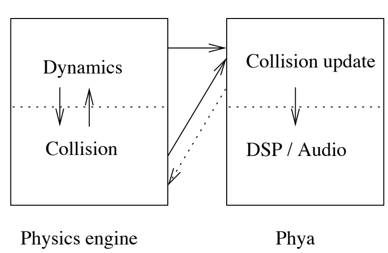

Figure 10: Phya system overview.

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.

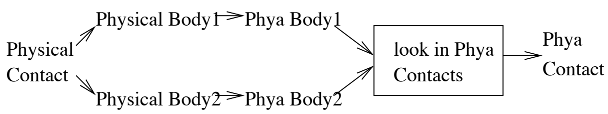

Figure 11: Find a Phya contact from a Physical contact.

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 neccessary 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 VIEW FROM INDUSTRY

There is little question among researchers that physically modeled audio of the kind considered here is of great value for creating virtual environments. More than this, the response from users is nearly always very positive. However the up take in industry has been disappointing, despite the ever increasing speed of computers and development of graphics and physics engines. Some tentative explanations are offered for this below, based on experience working with industry.

Computer game developers have the largest interest in this technology. Not all games are suited to physical audio, although an increasing number integrate physics engines of growing sophistication. Audio is usually considered of much lower priority than graphics, despite the fact that audio is frequently considered a very important aspect in reviews. Audio programming is often carried out by a non-specialist programmer just using sample playback libraries and programming interfaces for surround sound. There is often a natural resistance to acknowledge that out of house technologies could be valuable if they can not readily be reproduced in house. As we have seen, although a physical audio engine can be made robust, the connection with the physics engine controlling the movement of objects is much more fragile. Physics engines vary considerably in their operation, and are not designed from their outset to accomodate audio engines. In particular high quality continuous contact sounds require a quality of kinematic information beyond that which is visually acceptable. With ingenuity these problems can be surmounted in most cases, but they pose a significant strain on the developers resources. The problem is compounded because published research often focuses on a level of audio modeling detail that goes well beyond that required in a simulation, while not taking into account the practical requirements of a sound designer. In summary a significant shift in working practice is called for. Even among large developers who would be able to able to take a greater risk, there is a lack of evidence that progress is being made in physical audio. Several developers have blamed this on an increasingly risk-averse culture generally in the industry, and a lack of imagination among top management. The first to embrace new audio technologies and integrate with them will, however, surely gain an advantage on their competitors.

4. THE FUTURE

Phya is already well developed. Proposals for collaboration on virtual world projects are welcomed. It is currently being deployed in a virtual environment across the web, designed to explore synchronous virtual environments in the presence of latency. The result will serve as a comprehensive example with integration to an up to date publically available dynamics-collision engine, and hopefully encourage others to use it in their projects.

4.1. Extensions, cloth and water

Simulations of cloth and fluid surfaces, are becoming increasingly common in desktop physics engines. Suggestions are made here for appropriate models that can generate the associated sounds. Cloth generates a combination of sounds. When surfaces rub a continuous contact is sound is generated. This can be resonated and amplified by the surface if that is under tension. Similarly impacts can also occur between surfaces. The resonance can be modeled with a single modal resonator whose frequency is controlled by an average tension in the cloth. The excitation is a sum of the impacts and contacts occuring within the cloth. This requires access to detailed dynamic information that may not be readily availible.

Resonating bubbles have been identified as the dominant component in water/air sounds. Synthesizing stochastic populations of bubbles of different kinds can create a variety of realistic effects [5]. From a dynamic simulation of water surface flow we would hope to extract measures of turbulence, although this is an inherently difficult area for simulators to provide reliable information. These measures can then direct the populations statistics of the bubbles. It seems likely that the fluid simulation dynamics and stochastic process dynamics described in [5] will be insufficient to convincingly model the overall sound, because of the mismatch of scales. An intervening process could be introduced to manage the dynamical evolution of the population statistics in response to the raw turbulence measures.

When many bubbles are forming, it becomes increasingly hard to distinguish the resulting sound from spectrally evolving noise. This suggests that in these cases it would be better to use a noise model fed directly by the bubble population state, rather than summing individual bubbles, which is costly for large numbers.

Another possibility is to train the appended dynamics with sounds from real fluid flows, together with measurements of the macro dynamics. Using dynamical analysis similar to that which has been applied to musical instruments [15], a dynamic audio sound generator can be constructed. The stochastic nature of the fluid sound makes it more appropriate for this than precisely synchronized contact or instrument sounds.

5. CONCLUSION

The integration of physically modeled audio into a virtual environment is a complex task, and it is difficult to achieve high quality results. In the best examples, the audio generated integrates closely with the graphics in portraying the varied dynamic state of the system, and making the overall experience more realistic for the user. Sound designers are able to customize sounding objects according to their requirements, and ensure interactions are sonically interesting most of the time.

6. REFERENCES

[1] 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.

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

[3] 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.

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

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

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

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

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

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

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

[11] 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.

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

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

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

[15] B. Schoner, C. Cooper, and C. Douglas, "Data-driven modeling and synthesis of acoustical instruments," in *Proceedings International Computer Music Conference*, 1998.