Physically based sonic interaction synthesis for computer games

Nordahl, Rolf; Serafin, Stefania; Böttcher, Niels; Gelineck, Steven

Published in:
Audio Mostly 2006

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

Physically based sonic interaction synthesis for computer games

Rolf Nordahl, Stefania Serafin, Niels Böttcher and Steven Gelineck

Medialogy, Aalborg University Copenhagen
Lautrupvang 15
2750 Ballerup, DK
rn, sts, nboe05, sgel05@media.aau.dk

Abstract. In this paper we describe a platform in which sounds synthesized in real-time by using physical models are integrated in a multimodal environment. We focus in particular on sound effects created by actions of the player in the environment such as waking on different surfaces and hitting different objects. The sound effects are implemented as extensions to the real-time sound synthesis engine Max/MSP.[1] An 8-channel soundscape is spatialized using the vector based amplitude panning (VBAP) algorithm developed by VIlle Pulkki [17]. The sonic environment is connected through TCP/IP to Virtools.[2]

1 Introduction

In computer games and virtual environments, pre-recorded samples are commonly used to simulate sounds produced by the physical interactions of objects in the environment, as well as sounds produced when a user acts in the scenario by, for example, walking on different surfaces and hitting different materials. This approach has several disadvantages: first of all the sound designer needs to gather a lot of sonic material corresponding to the different actions and events in the environment. This is usually done by using sound effects libraries or recording sound effects, in the same way as it is done by a Foley artist in the movie industry [13]. Moreover, sampled sounds are repetitive, and do not capture the subtle nuances and variations which occur when objects interact with different forces, velocities, at different locations, and so on. This is usually overcome by applying processing to the recorded sounds, so some random variations are present.

However, by using sound synthesis by physical models these disadvantages can be overcome. Physical models are widely developed in the computer music community [19], where their main use has been the faithful simulation of existing musical instruments. One of the pioneers in the field of parametric sound effects for interactive applications such as computer games and virtual reality is Perry Cook. In his book [6], Cook describes several algorithms which allow to create synthesized musical instruments and sounding objects, mostly using physical principles. The issue of creating sound effects using synthetic models in order to syncronize soundtracks and animation was first explored in [20, 10] using a structure called *Timbre Tree*. Recently, synthetic sound models in computer animation have seen an increase of interest. Van den Doel et al. [12] propose modal synthesis [1] as an efficient yet accurate framework for the sonic simulation of interactions between different kinds of objects. The same synthesis technique has also been used by O'Brien et al. [16], as a computationally efficient alternative to the finite element based simulations proposed in [15]. Complex dynamical systems have also been simulated both sonically and visually by decomposing them into a multitude of interacting particles [3], in a system called CORDIS-ANIMA. In it, discrete mass-spring-damper systems interact with nonlinearities representing the input excitations.

In this paper, we describe a framework for real-time sound synthesis by physical models of different interactions in a computer game. We focus in particular on impact and friction sounds produced when a player interacts with objects in the environment. While the scenario's soundscape and the ambient sounds are created by using sampled sounds, the focus of this paper is on sounds produced by actions of the player. Examples are the sounds produced when the player hits hard objects or scrapes against surfaces of different materials with different forces and velocities. Such sounds are well suited to be simulated using physical models, especially given the fact that nowadays most game engines have physically based graphics engine in which forces and velocities of impacts and friction are calculated. Such physical parameters can be used as input parameters to the sound synthesis engine.

We are particularly interested in creating physically based sound models that are rich enough to convey information about a specific environment yet efficient to run in real-time and respond continuously to user or system control signals. In [9, 8], Gaver proposes a map of everyday sound producing events. Examples of basic level events might include hitting a solid, scraping it, explosions, and dripping noises. More complex events, then, can be understood in terms of combinations of basic-level ones, combinations which are structured in ways which add information to their simpler constituents.

Different platforms which allow to obtain sound synthesis by physical models are already available in the computer music community, although they have not yet been exploited in computer games. As an example, the Synthesis Toolkit (STK) by Perry Cook and Gary Scavone [5] is a collection of C++ classes which implement physical models of different musical instruments, mostly using the digital waveguides technique [19].

Another example is JASS (Java Audio Synthesis System) by

Kees van den Doel [12], a unit generator synthesis program written in JAVA, which implements physical models of different sound effects based mostly on modal synthesis [1].

The current development of novel interfaces for games, such as the Nintendo Wii,[3] stimulates the implementation of a tighter connection between gestures of the user and corresponding sounds produced [2]. This connection is strongly exploited in the computer music community, where so-called new interfaces for musical expression are developed to control several sound synthesis algorithm,[4] but it is yet not fully exploited in computer games and virtual reality applications. We believe that a stronger connection between player's gestures and resulting sonic environment can be obtained by using sound synthesis by physical models.

The paper is organized as follows. Section 2 introduces a multimodal architecture where sound synthesis by physical models have been integrated; Section 3 describes our strategies to track positions and actions of the user; Section 4 describes how the interactive sounds and the soundscape have been implemented; Section 5 introduces the visualization technique adopted, while Section 6 and 7 present an applications and conclusions and future perspectives respectively.

2 A multimodal architecture

Figure 1 shows a multimodal architecture in which sound synthesis by physical models has been integrated. The goal of this platform is to be able to precisely track positions and actions of the user, and map them to meaningful visual and auditory feedback. The position of the user is tracked by using a 3D magnetic tracker produced by Polhemus.[5]Moreover, a pair of sandals equipped with force sensitive resistors (FSRs) allow to detect when a user performs a step in the environment, together with the force of the impact. Such input parameters are mapped to the footsteps sounds which are synthesized using physical models. The Polhemus tracker is connected to the PC computer running Virtools, i.e., the visual rendering and game engine, while the footsteps controller is connected to the PC computer running Max/MSP. The two computers communicate together through TCP/IP. Finally, the synthesized interactive sounds, together with the ambient sounds are spatialised to an 8-channel surround sound system.

In the following, each component of the environment is described in more details. We start by describing the tracking systems used, since they are the input of the interactive sound designed.

3 Tracking the user

As mentioned above, the position and motion of the user are tracked in real-time using a Polhemus Fastrack tracker and an ad-hoc designed footsteps' controller.

3.1 The magnetic tracker

The Fastrack computes the position and orientation of a small receiver placed on top of a hat worn the user, as shown in Figure 4. This device provides six degrees of freedom measurement of position (X, Y, and Z Cartesian coordinates) and orientation (azimuth, elevation, and roll), which are mapped to the sound engine as described later. Given the limited range of the tracker of about 1 1/2 meter, the receiver was placed in the center of the 8-channels configuration.

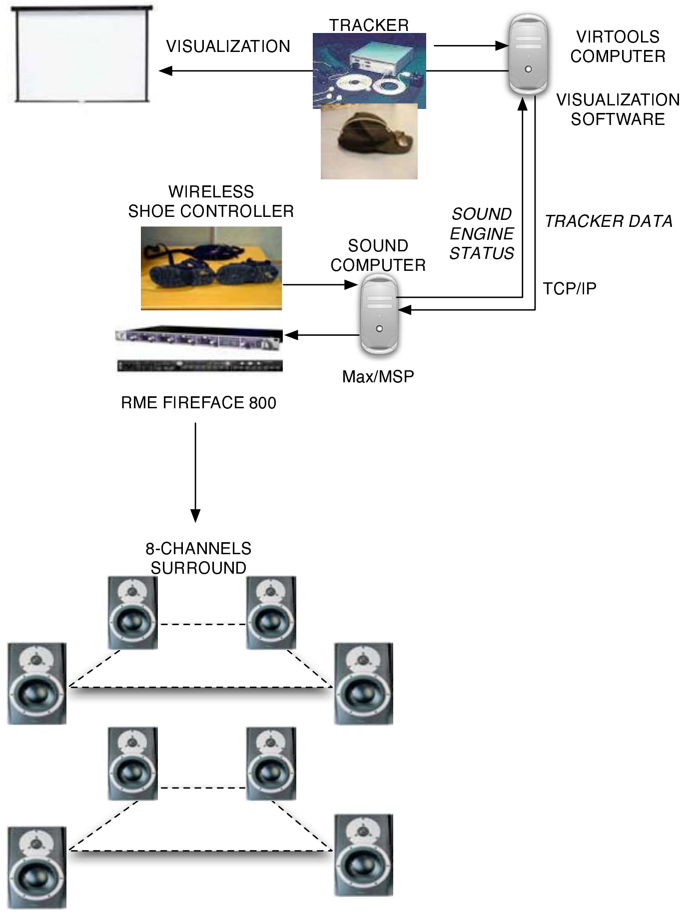

Figure 1: Connection of the different hardware and software components in the multimodal architecture. Two computers providing the visual and auditory rendering respectively communicate in real-time the tracker's data and the sound synthesis engine status.

3.2 The footsteps' controller

The users visiting the environment are asked to wear a pair of sandals embedded with pressure sensitive sensors, placed one in each heel as shown in Figure 2. Such sandals are wirelessly connected to a receiver, which communicates to the Max/MSP platform, by using ad ad-hoc designed interface [7].

Although sensing only the pressure of the impact on the floor does not allow to track all the parameters of a person walking in the environment, and more sophisticated footsteps' controllers have been built (see, for example, [11]), experiments with our configuration show that motion of subjects and sense of presence are significantly enhanced when self-sounds are added and controlled by this device [14].

[3] wii.nintendo.com/

[4] More information on this issue can be found in the proceedings of the New Interfaces for Musical Expression (NIME) conference, www.nime.org

[5] www.polhemus.com

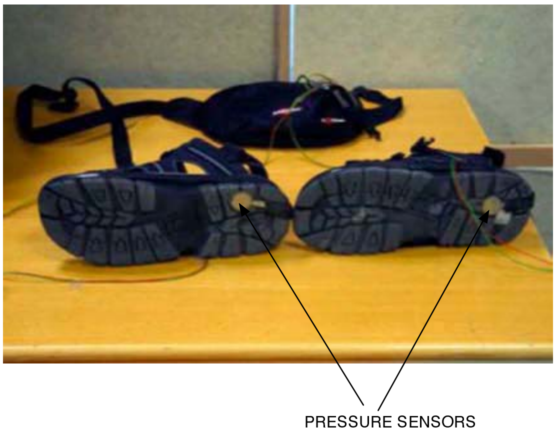

Figure 2: The interactive sandals are equipped with pressure sensors which trigger footstep sounds and forward movement in the virtual world.

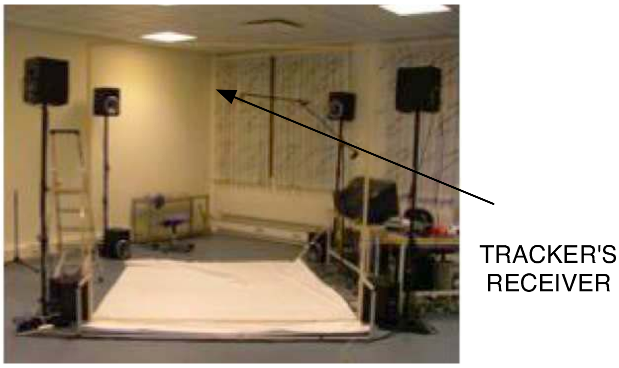

Figure 3: The setup of the 8 speaker system. The magnetic tracker emitter is situated in the center, directly above the user.

4 Sound design

Non speech sounds in computer games can be divided into soundscape or environmental sounds and sound effects. Soundscapes and environmental sounds are the typical sonic landmarks of an environment. They are usually reproduced by recording and manipulation of existing sounds, and do not strongly depend on the action of the users. On the other end, sound effects are usually produced by actions of the user in the environment, or by interaction between objects, and they can strongly depend on events in the environment. Such sounds are highly dynamic and vary drastically depending on the interactions and objects, and therefore are difficult to create in a pre-production process.

We decided to use sound synthesis by physical models for the creation of sound effects, and pre-recorded samples for the creation of the soundscape.

4.1 Interactive footsteps

Footsteps recorded on seven different surfaces were obtained from the Hollywood Edge Sound Effects library.[6] The surfaces used were metal, wood, grass, bricks, tiles, gravel and snow. Such surfaces were resynthesized using modal synthesis [1] and physically informed sonic models (PHISM) [6, 4]. The footsteps' synthesizer was implemented as an external objects in the Max/MSP platform.

[6] www.hollywoodedge.com

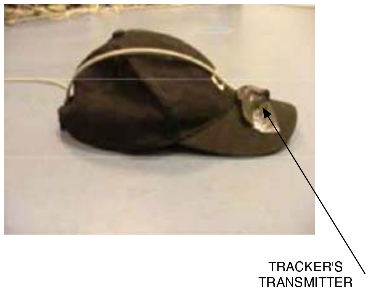

Figure 4: The Polhemus magnetic sensor is placed on the user's head, so auditory and visual feedback can be rendered according to the position and orientation of the user.

The control parameters of the synthetic footsteps were the fundamental frequency of each step and the amplitude and duration of each step. The amplitude and duration of each step were directly controlled by the users thanks to the pressure-sensitive equipped shoes. The sensors controlled the frequency of the steps, as well as their duration and amplitude. To enhance variations among different steps, the fundamental frequency of each step was varied randomly. The different surfaces varied according to the different scenarios of the game in which the user was present. As an example, when the user was navigating around a garden, the grass surface was synthesized, which became instantly a wood sound when the user was walking in a hardwood floor.

4.2 3D sound

The pre-designed soundscape which implemented ambient sounds was spatialized to an 8-channels system using the vector base amplitude panning technique (VBAP). VBAP is a method for positioning virtual sources to multiple loudspeakers developed by Ville Pulkki [17]. The number of loudspeakers can be varying and they can be placed in an arbitrary 2D or 3D positioning. In our situation, we chose a 3D configuration with 8 loudspeakers positioned in the vertexes of a cube, as shown in Figure 3. This is to preserve the same configurations as in CAVE systems.

The goal of the VBAP is to produce virtual sources which are positioned at a specific elevation and azimuth specified by the user. The idea behind VBAP is to extend the traditional panning techniques for two loudspeakers to a configuration of multiple speakers. We used the VBAP algorithm to position the ambient sound in a 3D space. Such sounds are pre-recorded samples which are positioned in a 3D space in real-time using the Max/MSP implementation of the VBAP algorithm. The algorithm allows also to simulate realistic moving sound sources, by continuously vary elevation and azimuth of the different input sounds.

5 Visual feedback

The visual feedback was delivered using a 2.5x2.5x2.5 m. single screen. 3D visualization was delivered using anaglyph and implemented in the Virtools platform. Virtools is a powerful game engine, which provides the possibility of having both block based programming in a similar way as Max/MSP, as well as implementation of ones own's blocks in C++. The 3D stereo was rendered using two Nvidia GeForce graphics cards [7]. A connection between Max/MSP and Virtools was obtained by using the flashserver object in Max/MSP[8] and the NSClient BB developed in Virtools.[9]

6 Application: an hide and seek game

In order to test the capabilities of the platform, an hide and seek game was developed. In this multi-users game the players have to find each others or escape from each others in a virtual environment. In the implemented example, the scenario is a small town. The idea behind the game is the connection between two VR CAVEs, with a user in each of them. The users are equipped with headset microphones, so they can communicate during the game. The sound of the other person is then panned into the exact position of the user in the came. By using auditory cues from the interactive sandals, one user can also derive location and position of the other person. The users are represented by avatars. They are only able to see the other user's avatar but their own. Two persons outside the environment are connected to the game via LAN network. The can communicate with the users inside the game, and their goal is to transmit information about the location of the opponent. The external users are also able to upload 3D objects or sounds in the game in real-time. In this way, they are able to disturb the opponent user and enhanced the atmosphere by varying the current soundscape.

Figure 5: The view of one user playing the hide and seek game.

[7]www.nvidia.com

[8]The flashserver object for Max/MSP was developed by Olaf Matthes.

[9]The NSClient BB was developed by Smilen Dimitrov at Aalborg University in Copenhagen

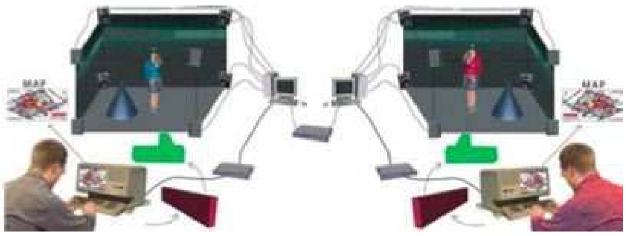

Figure 6: Setup of the game with four users.

7 Conclusion

In this paper we have described a multimodal architecture where interactive sounds synthesized by physical models as well as ambient soundscapes have been integrated.

As done in [12] and [18], our current focus is on impact and friction sounds produced by actions of the user while interacting in the environment. In particular, we have focused our description on the use of footsteps sounds, since they play an important role in game design. We are currently extending this architecture to the use of action sounds produced by interaction of the user with other body parts, such as sound produced when the user hits, grabs and touch objects in the environment. As mentioned in the introduction, computer games currently released in the market use sampled sounds instead of computer generated sounds. The main reason for this choice is from one side the high computational cost of producing high fidelity sound synthesis by physical models, but on the other side the lack of sound quality of most synthesized sounds. Even in the field of musical instruments, which have been synthesized by using physical models for more than three decades, the quality of physical models is yet not as high as the original instrument which they are trying to simulate. Of course many progress has been done in this area, but we are not yet at a point where physical models can be used in a commercial applications.

We are currently conducting experiments to understand if the use of physically based sounds enhances realism and quality of the interaction in a game.

References

[1] J.M. Adrien. The missing link: Modal synthesis. In *Representations of Musical Signals.* in: G. De Poli, A. Picalli, and C. Roads, eds., MIT press, 1991.

[2] T. Blaine. The convergence of alternate controllers and musical interfaces in interactive entertainment. In *Proc. International Conference on New Interfaces for Musical Expression (NIME05)*, 2005.

[3] C. Cadoz, A. Luciani, and J.-L. Florens. Physical models for music and animated image. The use of CORDIS-ANIMA in Esquisses: a Music film by Acroe. In *Proc. Int. Computer Music Conf.*, Aarhus, Denmark, Sept. 1994.

[4] P. Cook. Physically informed sonic models (phism): Synthesis of percussive sounds. *Computer Music Journal*, 21(3):38–49, 1997.

[5] Perry R. Cook. Toolkit in C++, version 1.0. In *SIGGRAPH Proceedings*. Assoc. Comp. Mach., May 1996.

[6] P.R. Cook. *Real Sound Synthesis for Interactive Applications*. AK Peters, Ltd. Natick, MA, USA, 2002.

[7] S. Dimitrov and S. Serafin. A simple practical approach to a wireless data acquisition board. In *Proc. NIME*, 2005.

[8] W. Gaver. How do we hear in the world?: Explorations in ecological acoustics. *Ecological psychology*, 5(4):285–313, 1993.

[9] W. Gaver. What in the world do we hear?: An ecological approach to auditory event perception. *Ecological psychology*, 5(1):1–29, 1993.

[10] J. Hahn, J. Geigel, J. Lee, L. Gritz, T. Takala, and S. Mishra. An Integrated Approach to Audio and Motion. *Journal of Visualization and Computer Animation*, 6(2):109–129, 1995.

[11] A. Benbasat Z. Teegarden J. Paradiso, K. Hsiao. Design and implementation of expressive footwear. *ICM Systems Journal*, 39(3):511–529, 2000.

[12] P. Kry K.van den Doel and D. Pai. Foleyautomatic: Physically-based sound effects for interactive simulation and animation. In *Proc. ACM SIGGRAPH*, 2001.

[13] R.L. Mott. *Sound Effects: Radio, TV, and Film*. Focal Press, 1990.

[14] R. Nordahl. Design and evaluation of a multimodal footsteps controller with vr applications. In *Proc. Enactive*, 2005.

[15] J. O'Brien, P. R. Cook, and G. Essl. Synthesizing Sounds from Physically Based Motion. In *Proc. Siggraph, Computer Graphics Proceedings*, pages 529–536, 2001.

[16] J. O'Brien, C. Shen, and C. Gatchalian. Synthesizing Sounds from Rigid-Body Simulations. In *Proc. Siggraph, Computer Graphics Proceedings*, pages 175–203, 2002.

[17] V. Pulkki. Generic panning tools for max/msp. In *Proc. ICMC*, 2000.

[18] D. Rocchesso. Physically-based sounding objects, as we develop them today. *Journal of New Music Research*, 33(3):305–313, 2004.

[19] J.O. Smith. Physical modeling using digital waveguides. *Computer Music Journal*, 16(4):74–91, 1992.

[20] T. Takala and J. Hahn. Sound Rendering. In *Proc. Siggraph*, pages 211–220, 1993.