Audio Engineering Society

Convention Paper

Presented at the 128th Convention
2010 May 22–25 London, UK

FoleySonic: Placing Sounds on a Timeline Through Gestures

David Black[1], Kristian Gohlke[1], and Jörn Loviscach[2]

[1] University of Applied Sciences, Bremen, 28199 Germany
dblack@mevis.fraunhofer.de, kgohlke@acm.org

[2] University of Applied Sciences, Bielefeld, 33602 Germany
joern.loviscach@fh-bielefeld.de

ABSTRACT

The task of sound placement on video timelines is usually a time-consuming process that requires the sound designer or foley artist to carefully calibrate the position and length of each sound sample. For novice and home video producers, friendlier and more entertaining input methods are needed. We demonstrate a novel approach that harnesses the motion-sensing capabilities of readily available input devices, such as the Nintendo Wii Remote or modern smart phones, to provide intuitive and fluid arrangement of samples on a timeline. Users can watch a video while simultaneously adding sound effects, providing a near real-time workflow. The system leverages the user's motor skills for enhanced expressiveness and provides a satisfying experience while accelerating the process.

1. INTRODUCTION

Faster network access and the availability of cheap processing power on the consumer market have been the breeding ground for online video sharing services such as YouTube and Vimeo. These tools allow the consumer to produce and publish their own audio-visual content online.

Even though means for acquiring and editing sound and visual material are widespread, homemade productions of these often rather short video clips frequently lack expressive high-quality sound effects. The sound tracks may simply consist of a single track of music or of the unedited audio that was recorded and cut along with the video. The approach presented here focuses on providing a tool for the casual producer to intuitively work with foley sounds and create a more expressive soundtrack using readily available low-cost motion-sensing devices.

Suggestions are given that may lead to fruitful integration with both consumer and professional workstation software. Further, a better user experience for interacting with such tools could entice users to devote more

efforts to creating better foley soundtracks and enhance the overall quality of casual video productions.

1.1. Sound Parameter Editing in the Audio Workstation

Most present consumer solutions for audio editing for video include editing workflows which demand that the user directly accesses abstract low-level parameters of sound, such as reverb decay time, low-pass filter frequency, sample loop start time, and oscillator fine tuning percentage. These values are often represented in software by means of on-screen envelope graphs, which depict the changes in a value over time corresponding to a timeline and displayed alongside waveforms and video clip information. Values may be entered using the mouse and keyboard or recorded in real time using any number of hardware controls, such as knobs, faders, and xy touch pads. Although these envelopes allow for a seemingly great deal of control and precision when performing sound design tasks, this method suffers from a number of deficiencies.

First, this approach only provides a rather technical and abstract way of interacting with the sound, namely by adjusting each numerical parameter value individually. Thus, it lacks the expressiveness and directness often required when creating foley sound events and placing them on the timeline. Since the auditory results of modified values must constantly be auditioned after they are entered, disengagement arises between the act of entering a value and evaluating its resultant sound.

Second, manually entering sound parameters is often time-consuming, as it requires repetitive interaction. When using this approach, only one parameter may be edited at a time, forcing the editor to continuously switch between different editing parameters in order to create a harmony in which all parameters work together to build a certain foley sound. The editor must first estimate a value to enter into the system and then audition its result, therefore creating much guesswork, in particular for a layperson.

Third, this approach limits the ability of the editor to react to movie events directly while editing the sound. The editor must often pause the playback and resort to frame-level editing to calculate when a certain foley sound event should occur, and then continue by entering envelope or other parameters. Finally, this approach limits its usefulness to editing in a studio. Live performances and theater use are precluded because they require on-the-fly parameter adjustment of several parameters at the same time. Even when using common hardware controls such as MIDI controllers, the previously mentioned limitations found in standard audio editing suites often lead to unsatisfactory user experience and limited control.

1.2. Concepts from Professional Foley

In contrast to consumer-level foley editing, professional foley artistry has long used motion and physical objects to create expressive effects tracks. The “instruments” a foley artist uses to create and record sounds from can be made up of almost any object that can generate sound, such as shoes for footsteps, coconuts for horse clopping, and plastic foil for frying eggs. Such props are often simultaneously recorded while a particular scene of a movie is being played back. The benefit of professional foley techniques is that they provide a high level of expression and flexibility as they allow the foley artist to directly react to the video material. However, this approach demands that the foley artist is highly trained. In addition, after foley effects are recorded, parameters can only be edited within certain boundaries, and sometimes new takes must be recorded to achieve the desired sound.

2. RELATED WORK

Motion gestures have previously been used for digital audio production in different contexts. A brief overview of the approaches that are most relevant within the scope of this paper are given in the following.

Kyma X is an environment for professional sound design on the computer which can be controlled using the Wii Remote [10][11] and various other alternative controllers, such as graphic tablets.

The Junxion software package [17], developed at the STEIM Center, and the Osculator [15] enable the use of different physical devices such as USB joysticks, MIDI controllers, the Wii remote, and various sensor platforms as audio controllers. The software is useful for conditioning and sending out both MIDI and OSC data. The GlovePIE ‘Programmable Input Emulator’ software [8] provides similar control functionality and allows creating complex control mappings using a dedicated scripting language. However, such software is mainly useful to map control events rather than to provide a comprehensive way to generate sound events or to allow

bidirectional transport control within the digital audio workstation (DAW).

The Wii Remote has also been used as a musical instrument in various contexts as a tool for musical improvisation [19] or as a virtual shaker [9] based on a physical model. The use of shaker controllers as input devices for the PhiSEM sound engine has been demonstrated [16]. The shaker controllers that were custom fitted for this purpose provided an easy to grasp instrument that was well received as they offered a good user experience.

Synthesis of sound effects for computer games using a Wii Remote in conjunction with techniques for physical modeling has also been demonstrated [3]. The use of different three-dimensional motion gestures that loosely mimic the physical motion of interacting with different musical instruments has been evaluated [4]. The Scrubber controller [6] enables the creation of a variety of friction-induced sounds based on different synthesis models.

3. THE FOLEYSONIC

To overcome the limited expressiveness of creating foley soundtracks using manually entered sound parameters, and to provide an additional tool for professional foley artists, we demonstrate a solution that leverages the expressive capabilities offered by commercially available motion-sensing devices, combining the best features of both the traditional foley workflow and manual tweaking of sound parameters.

Traditional foley recording is based on the manipulation of physical objects in such a way that sounds are generated, which can be recorded and placed onto a timeline. We take this approach and apply it to the motion-sensing device in a way that reflects the workflow of the traditional foley artist. Just as the foley artist might simply shake a large piece of paper to generate a sound to mimic (e.g., thunder), the FoleySonic user might shake the motion-tracking device in the same manner to change the sound design parameters of a virtual thunder instrument.

The core concept is built on the notion that traditional motions used in creating sound with physical objects can be used to control parameters of virtual sound devices and audio tracks to give the user effortless enhanced expressivity and control of foley sound design characteristics.

The current version of the FoleySonic employs a Wii Remote for motion tracking, which contains a three-axis accelerometer and is connected to the Motion Plus accessory [20] that contains an additional three-axis gyroscope in order to reliably detect the rotation of the device. This setup allows controlling six degrees of freedom (DOF), as shown in Figure 1. However, the system is not restricted to the Wii Remote and could be used with any other 6-DOF tracking device that is able to connect to a computer.

The current FoleySonic prototype was developed in order to evaluate the interaction concept. It uses available software wherever this was feasible. The system consists of two main components.

The first component receives and processes incoming motion-tracking data from the Wii Remote, extracting parameters to be used in sound control and forwards them as OSC [7] messages. To achieve this, we employ the Osculator software [15] mentioned before. The software also carries out some basic filtering of the raw sensor values in order to smooth values and minimize value drift. However, for the purpose of the foley sonic this is negligible as the user can instantly recalibrate the system by pressing the ‘home’ button the Wii Remote.

The second component consists of a framework of templates that interpret the incoming OSC data stream and map them to sound parameters and sound event generators on different tracks of a DAW. For this purpose we have developed a collection of Max/MSP patches that allow for complex value mappings. The software patches run in the background and translate each incoming OSC event or combination of multiple events into a carefully concerted stream of different MIDI messages, which are then used to control several audio parameters inside a software DAW simultaneously in a meaningful way as described further below. The final component also maps certain OSC events to enable direct control of transport and other ancillary functions of the audio workstation such as play, record, track select, and punch point selection.

In order to be useful, even in a prototype state, the FoleySonic should be integrated into the existing studio workflow. Therefore, it is not a dedicated audio editing suite on its own but rather enables control of various existing DAW software. In order to use the system, the user simply needs to load a template project file into the DAW software. This project is preconfigured to work with the FoleySonic MIDI messages and record the

incoming events on multiple tracks. Furthermore, the template project also provides a set of different foley sounds that have been created and evaluated in previous work [1]. This collection was specifically designed to cover a large variety of foley events. The standard sounds for each foley preset are loaded into dedicated software samplers or other sound generators within the DAW, which are then controlled by the FoleySonic system. The control events generated by the FoleySonic are used to both trigger playback and control multiple parameters of the samplers. All sounds can be swapped by the user.

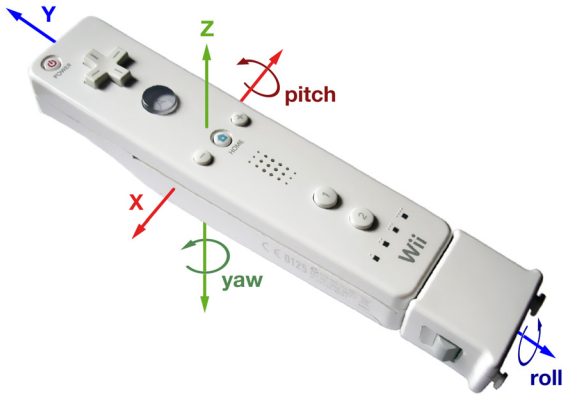

Figure 1: A Wii Remote with the attached gyroscope accessory that enables the device to track six degrees of freedom.

3.1. Mind Models for Foley Sounds

The FoleySonic software includes a number of mapping presets that seek to provide a meaningful relationship between the user's motion input and the sound generator. Each of these presets makes use of a certain mind model that is loosely related to interacting with physical sound-producing devices, although the presets do not necessarily have to mirror the physical model of the controlled sounds. Instead, they are rather designed to provide simple modalities to easily produce different types of sound events and yield a satisfying user experience. The template project file contains dedicated tracks that are used to record each of the different sound categories. The preset categories are described in the following.

3.1.1. One-Shot Trigger

This template uses a button on the Wii Remote to trigger and tweak a single one-shot sample. Turning the device around the X axis, simultaneously affects volume and reverberation. Yaw is used to control stereo panning. This allows creating basic 2D spatialisation effects.

An example application of this model may be a gunfight between two opponents. The preset can be used to create sound for a scene in which two fighters are shooting at each other from different sides of the screen; they might also be located at different distances to the camera. To create foley sounds for this scene using the FoleySonic, the device is pointed to one side of the room for the first fighter's gun sound and then pointed to the other side of the room for the opponent's sound. The pitch angle of the device controls the perceived proximity of a sound. Pressing the trigger button on the device (i.e. the large 'B'-button on the bottom of the Wii Remote) sends a note-on event, which in turn triggers the software sampler holding the gunshot sound.

In addition, rolling the device around the Z axis adjusts the overall volume output volume of the sampler; this gesture is similar to turning a rotary knob on a hardware controller. Panning and volume parameters are automatically recorded as envelopes on the audio track. Any effect controls are also recorded as automation data for subsequent optional fine-tuning.

3.1.2. Velocity-based Trigger

In this template, sound events can be generated by the user using a striking, whipping motion with the device downward along the Z axis. This mimics the motion of hitting a drum, shaking maracas, or slapping a table. Using this means of input, sound events may be triggered by moving the device sufficiently quickly in opposite directions for note events to be generated at each downstroke. The velocity of the device at the trigger instant is then used to control the volume and the audio compression of the resulting sound event. Thus, quiet events may be generated with slower movements, and louder events may be generated with faster movements, much as with physical drums and shakers. As in the previous preset, yaw and pitch angles are again mapped to volume panning and reverb mix. The roll angle again controls the overall volume.

With such a template, for instance, a series of footsteps or sounds for a fistfight may be recorded. A scene may require that a character walks across the screen, from left to right, entering from afar down a large hallway. To realize this using FoleySonic, the user would make light movements with the device, gradually increasing speed, and thus the intensity of the sound, until the character appears on screen. Reverb and panning can be simultaneously modified to mimic motion down a large hallway.

3.1.3. Velocity Trigger with Modulating Release

Often, sound events must be triggered but also be allowed to sustained and modulated for a variable amount of time, unlike discrete footsteps and gunshot sounds. For such a purpose, we provide a template that allows sounds to be triggered and sustained as long as the user desires while maintaining basic control over the sound.

Thunder sounds for a storm scene may be generated in such a way. The user may first "strike" the virtual space with the device in an expressive upward or downward gesture, much as with the aforementioned footstep motion. When the user presses and holds a button on the device the sound enters a sustain loop that is repeated until the button is released again. While being in the sustain loop the user can control the swelling of volume and some soft distortion or a delay effect to create an individualized tail of sound for each thunder event. Intense shaking of the device also "fattens" the resulting sound by adding equalization and compression. This preset can be used for any sound that starts off with a louder event followed by a tail of arbitrary length.

3.1.4. Intensity and Rhythm

In this template, the aggregate motion of the motion-sensing device is mapped to the intensity and repetition rate of individual sound events that slightly change over time.

This can be used, for instance, to model the sound of ambient noises, such as rush-hour traffic, a jar of coins being shaken or a swarm of bees humming. The more accumulated motion the device is given, the more intensely and frequently the virtual sound device generates polyphonic sound events. Sound is only played back while the main trigger button on the Wii Remote is held down. Releasing the button causes the sound to fade out within a third of a second, enabling soft interruptions of the generated sound through intuitive control of the volume envelope. Pressing the trigger again quickly fades in the sound. Because mapping direction may not be of utmost importance for ambient sounds, the user may focus entirely on modulating the intensity and the rhythm of the sound events.

3.1.5. Control of Continuous Sound

As opposed to the mind models of the templates mentioned before, which employ trigger-based sound generation, we also provide means of controlling continuous or looping sound generators. In these cases, parameters from the motion-tracking device may be mapped to the sound generator based on logical methods of controlling the desired physical object.

One example might be that of a car engine. A changing pitch angle of the Wii Remote models the accelerator of the car, thereby raising or lowering the audible RPM of the engine when the angle of the device is changed. Thus it reflects the interaction with the acceleration controls of a car, a boat, or an airplane. Again, audio panning is controlled by the yaw angle. However, unlike most of the previous presets, rolling the device does not only control the overall volume, but also adjusts a reverb mix, as the pitch angle is already used for controlling parameters more central to this sound category. As opposed to the current sample based approach, a further extension of this model may also provide simplified control over more complex sound generation techniques, such as synthesizers that may rely on many interlaced parameters to fully realize a particular sound.

3.1.6. Pseudo-Physical Models

Often, the physical manipulation of an actual everyday object can be employed as a metaphor for the interaction with the motion-sensing device.

The current prototype uses such a metaphor for creating the creaking sound of a wooden door, including the sound of the door hitting the frame. For this purpose, the Wii Remote can be placed perpendicular to a surface, with the Z axis facing upwards and the bottom of the device touching the surface. If the user adjusts the pitch angle between of the device relative to the surface, the sound of a creaking door is played back in such a way that it relates to the current angle and the speed of motion, mirroring the audible properties one would expect from a door. Once the device reaches the horizontal plane at sufficient speed, an additional sound is being played that simulates the sound of the door slamming

into frame at a level of intensity that relates to the speed of impact.

In the current prototype, this model is also used to scrub through a given sample at varying speed, while simultaneously adjusting some effect parameters, such as equalization, reverb and distortion mix according to the speed of motion. Rolling the device around the X axis changes the simulated friction of the resulting sound, allowing for intuitive and expressive playback of a single sample. This sound category could be further enhanced by employing sophisticated physical models [3] in order to easily achieve a desired sound from a complex sound generator.

Furthermore, the model can be used to control a variety of other friction- or tension-based sounds, such as the sounds of a creaking mast on a sailing boat, a turning key, windshield wipers or the sound of objects being dragged over a surface.

3.2. Control of General Track Parameters

In addition to controlling the sound output of devices such as samplers and other virtual instruments, the FoleySonic provides a template that does not generate sound by itself, but directly records the input from the Wii Remote as automation data, which can then be used for the expressive modification of automation parameters in all audio tracks, and thereby also provides more expressive control over the sound. In many cases, this may be as simple as controlling parameters such as panning, volume, effect levels, or other commonly used automation values. However, this method requires more configuration effort from the user to map the input to the desired parameters.

3.2.1. Transport Control and Selection

In addition to providing a framework of templates for control of sound generating devices using foley concepts, the motion-tracking device is also used to facilitate transport control of the audio workstation and for toggling various basic functions. The device can be used as a general remote control for basic controls of the DAW. Common controls used during foley sound editing are implemented, such as record, pause, play, fast-forward, rewind, punch in and out points, and track record, mute, solo and select. Although this might not allow total freedom from the traditional audio workstation controls, the user may more easily concentrate on the foley task rather than frequently switching between different input modalities, such as the mouse, the keyboard, various hardware controllers or the motion-tracking device. The directional pad control further functions as a shortcut to switch between different samples within each of the given the foley categories.

4. CONCLUSIONS

We have demonstrated a prototype system that enables creating an expressive foley soundtrack and is aimed at enhancing the user experience and the overall workflow of foley artists. The system can be adapted to work with a variety of legacy DAWs by using custom project files and enables recording of automation parameters that may be later edited to achieve desired accuracy in foley sound design tasks. Informal pilot evaluations and interviews with prospective users have shown that the FoleySonic tool is an entertaining and powerful approach to foley sound placement on the timeline. Users were drawn to the intuitive modeling of foley sound design parameters to the motion tracking device and were eager to experiment and record tracks of parameters onto the timeline in their audio workstation. Much time was spent by users interacting with all of the possible modes of operation, exploring the relationships between physical motion and sound. In this way, users demonstrated a great range of immediate expressiveness that could hardly be achieved when using only a mouse and standard MIDI controllers. The ability of the tool to operate the DAW as a remote control was also seen favorably by users.

The primary limitation of the system is the remaining necessity to manually assign some parameter mapping from the system to the audio workstation via MIDI messages, despite the mappings that are already present in the included template files. In our evaluation prototype, we used Ableton Live as DAW, which allows some of these limitations to be overcome by employing "user remote scripts" that enable a better integration of control into the software, although Live still requires a few parameters to be manually mapped.

Even though our approach is designed to accommodate a wide range of different foley events, no amount of included templates, samples and suggested mind models may cover the entire interaction space between the motion tracking device and the sound generator. Therefore, the user may occasionally be required to tweak the provided templates or create entirely new ones to satisfy certain individual foley sound conceptualizations. However, the system can provide a good introduction to

making an expressive foley soundtrack and function as a responsive recorder for creative ideas, as it enables the user to quickly sketch out the structure of a foley track that can be modified in detail later. More in-depth testing with a wider range of prospective users will help us determine which of the preset templates are most useful and which may need to be changed, merged or removed.

Ongoing work addresses a better integration of the system with various commonly used audio workstations. However, a thorough integration would require dedicated programming APIs for interfacing with the software. With commercial software, such APIs are scarce and, if available, they are mainly based on proprietary technology or protocols [5] or provide only limited functionality [18][14]. Ultimately, manufacturers of audio editing software should be encouraged to support open standards and embrace community efforts, such as user-contributed APIs [12], as they can enable a better interactive experience when being used with their products.

5. REFERENCES

[1] Ableton Live, http://www.ableton.com/live (last accessed: March 5, 2010).

[2] Black, D., Heise, S., and Loviscach, L. Generic Sound Effects to Aid in Audio Retrieval, AES Paper 7795 (126th AES Convention, 2009).

[3] Böttcher, N., and Serafin, S. Design and Evaluation of Physically Inspired Models of Sound Effects in Computer Games. Proceedings of the 35th International AES Conference: Audio for Games.

[4] Bott, J. N., Crowley, J. G., and LaViola, J. J. 2009. Exploring 3D Gestural Interfaces for Music Creation in Video Games. Proceedings of the 4th international Conference on Foundations of Digital Games. FDG 2009. pp. 18-25.

[5] Boyer, R., Campbell, P., Freshour, S., Kloiber, M., McTigue, J., Milne, S. EuCon: An Object-Oriented Protocol for Connecting Control Surfaces to Software Applications. AES Paper 6957 (121st AES Convention, 2006).

[6] Essl, G., O'Modhrain, S. Scrubber: an interface for friction-induced sounds. Proceedings of the 2005 Conference on New Interfaces For Musical Expression, NIME. pp. 70-75.

[7] Freed, A., Schmeder, A. Features and Future of Open Sound Control Version 1.1 for NIME. Proceedings of the 2009 Conference on New Instruments for Musical Expression, NIME.

[8] GlovePIE, http://glovepie.org/ (last accessed: March 5, 2010)

[9] Heise, S., and Loviscach, J., A Versatile Expressive Percussion Instrument with Game Technology, Proceedings of IEEE ICME 2008. pp. 393-396.

[10] Kyma, http://www.symbolicsound.com/cgi-bin/bin/view/Learn/UsingTheNintendoWiimoteWithKyma (last accessed: March 5, 2010).

[11] Lee, J. C. "Hacking the Nintendo Wii Remote." IEEE Pervasive Computing. pp. 39-45.

[12] Live API, http://code.google.com/p/liveapi/ (last accessed: March 5, 2010).

[13] Max/MSP, http://cycling74.com/products/maxmspjitter/ (last accessed: March 5, 2010).

[14] Max for Live, http://www.ableton.com/maxforlive (last accessed: March 5, 2010).

[15] Osculator, http://www.osculator.net/ (last accessed: March 5, 2010).

[16] Shaker Controllers to Control PhISEM http://soundlab.cs.princeton.edu/research/controllers/shakers/ (last accessed: March 5, 2010).

[17] Steim Junxion, http://www.steim.org/steim/junxion_v4.html (last accessed: March 5, 2010).

[18] Steinberg, VST Module Architecture http://ygrabit.steinberg.de/~ygrabit/public_html/vstsdk/OnlineDoc/VST_Module_Architecture_SDK/index.html (last accessed: March 5, 2010).

[19] Toyoda, S. Sensillum: An Improvisational Approach to Composition. Proceedings of the 2007 Conference on New Interfaces for Musical Expression, NIME. pp. 254-255.

[20] Nintendo Wii Motion Plus http://www.nintendo.com/wii/what/accessories/wiimotionplus (last accessed: March 5, 2010).