GSoundAU

Sound Propagation for Professional Audio Production

Matthew Adams

2016

1 Introduction

From echos used for making pop ballads sound full and rich to carefully-manipulated recordings of actors meant to immerse audiences in a film, professional audio productions go to great lengths to mimic the ways sound changes when it bounces around a room. Although we don't generally think about it, humans have built up intuition about these propagation effects; even if they couldn't explain why, people know that talking in an office cubicle sounds very different from talking in a stone cathedral.

Until now, sound designers and mixers have mimicked these perceptual effects of sound propagation using a variety of techniques. Unfortunately, these are generally either inaccurate approximations (like digital reverberation filters) or expensive to produce and for limited applications (e.g. recording samples of every orchestral instrument for many different listening positions in a famous concert hall). For audio professionals, accurate sound propagation effects would mean more realistic results achieved more quickly and easily.

To that end, we have created GSoundAU: a system that seamlessly integrates fast and accurate sound propagation with the core tools of modern audio production. GSoundAU empowers users to simulate the sound of their audio propagating through any 3D environment model in real time. GSoundAU's intuitive interface, inspired by visualization research, makes it easy to effectively stand anywhere in a virtual room and hear your audio bouncing around-making for much more flexible and accurate sound propagation effects than can be achieved with most current systems.

The rest of this paper is structured as follows. Section 2 offers a quick overview of the current state of sound propagation effect tools and techniques in modern audio production. Section 3 describes GSound, the sound propagation engine chosen to power GSoundAU. In section 4, we present the GSoundAU user interface design and development in depth, and then section 5 concludes the paper and describes future work.

2 Overview of Current Tools and Techniques

2.1 Digital Audio Workstations

The centerpiece of modern audio production is a software platform called a digital audio workstation (DAW). DAWs such Avid Technology's Pro Tools[1], Apple's Logic Pro[2], and Adobe Systems' Audition[3] facilitate every stage of the sound production process- from recording and synthesizing to mixing and mastering- for music, film, and voice-over work. The systems vary but they are all built around the basic task of recording multiple tracks of audio and moving them on a timeline to choose when they play relative to each other, as shown in Figure 1. Beyond that, each DAW uses a wide selection of plug-ins- both built-in and third party- to do the heavy lifting of modifying the audio. These plug-ins are virtually connected in series, so that altered audio flows out of one and into another like the effect circuitry boxes of DAWs' analog predecessors. For example, a mixing engineer might commonly use an equalizer plug-in to adjust the levels of various frequency ranges, then send the modified signal through a compression plug-in to smooth out the dynamic range of an audio signal, as shown in Figure 2.

[1] http://www.avid.com/pro-tools
[2] http://www.apple.com/logic-pro/
[3] http://www.adobe.com/products/audition.html

Figure 1: Apple's Logic Pro DAW's main view uses a common timeline layout, with individual audio tracks, each containing audio clips arranged in time, layered together to produce the composite output. Effects are applied on a per-track basis using plugins (not pictured) and affect any audio clips on a track's timeline.

DAWs are so central to the standard audio engineering workflow, that bringing accurate sound propagation to professionals means building a plug-in. Plus, having a fully-integrated system enables us to leverage the powerful parameter animation tools DAWs provide. In designing this system, we have elected to develop an audio unit (AU) plug-in, the format built primarily for Apple's Logic Pro DAW. Besides simply being the DAW we are most familiar with for creative endeavors, Logic offers a couple advantages for this work:

- AU plug-ins also are natively supported by Apple's popular consumer-level DAW, Garageband, so a broader audience can easily explore accurate sound propagation.
- While not the market leader, Logic Pro is a well-established, high-traction DAW for professional music creation that has been used in the creation of major pop hits [1].

Figure 2: Plugins get added to individual channel strips in series. This channel's audio gets sent through an equalization plugin (for frequency manipulation) then a compressor plugin (to limit dynamic range).

2.2 Current State of Sound Propagation Effects

Currently, sound propagation effects fall into two broad categories: reverberation effects and recording audio that includes the propagation effects of a specific room. Reverberation, or reverb, is the sound produced by the hundreds of echos bouncing off every surface of an environment. A wide variety of reverb effects have been developed over the years, ranging from analog approaches where the input sound would vibrate physical plates and springs to digital effects, which use various digital signal processing techniques or even simulate the analog vibrations. These types of reverb effects, in concert with frequency manipulation from equalizers, shine when used as a means of adding echos to make audio sound more detailed or full, but they are limited when the goal is to create realistic reverberation for a specific environment. For example, films and television often cannot use on-set audio for a variety of reasons and end up overlaying the audio in studio, in a process called automatic dialogue replacement (ADR). Skilled ADR engineers have to painstakingly dial in various reverb parameters until the dialogue sounds like it fits in the physical scene, but the parameters being adjusted are not directly related to the actual environment. Some reverbs, like Wave Arts' Panorama plug-in mitigate this challenge by focusing on localization parameters [2]. However, even these specialized reverbs are fundamentally not designed to be able to reproduce the more complex effects that arise from sound sources that can be temporarily occluded or move quickly enough to have their pitch shifted by the Doppler effect.

When practical, the most straightforward method to increase realism is, of course, simply recording sound in the desired environment with natural reverberation. Unfortunately, this approach gets costly quickly, if feasible at all, and doesn't easily allow for tweaking the reverb after the recording stage of production has been completed. One solution on the market is collections of recordings that have been sampled from instruments in high-quality concert halls or studios, but these collections only solve the problem for specific use cases- commonly, creating a generic large orchestra. To generalize the sampling solution, some plug-ins, such as Universal Audio's Ocean Way Studios plug-in or Silver Spike's RoomMachine 844, sample the impulse response (echoes from a burst of sound) of various positions in a specific environment and convolve that response with the user's source audio [3, 4]. Such plug-ins are effective for emulating the spaces they sample and model but, as they rely on sampling, cannot be used for generic environments.

Clearly, simulated sound propagation opens up a new world of possibilities, and there are already a few plug-ins that use simulation results. Some, like Waves' NX, focus on a single environment and are arguably effectively cheaper-to-produce versions of convolution-based reverb plug-ins [5]. QuikQuak's RaySpace stands out because it simulates sound propagation for generic spaces and can even compute and export impulse responses for use in convolution reverbs [6]. RaySpace uses a proprietary source-to-listener ray tracing algorithm the authors call Diffuse Bubble Tracing to simulate bounce paths and calculate reverb [6]. RaySpace is a solid foundation for sound propagation plug-in efforts, but the underlying algorithm does not appear to incorporate diffusion or diffraction effects and its efficiency (as judged by latency) should be improved to make using the plug-in more practical.

3 GSound

3.1 Choosing a Sound Propagation Algorithm

The fact that propagation effects need to respond in real time significantly restricts the type of approaches that can be used. Wave-based approaches, which simulate the physical waves of air pressure that constitute sound, are the most realistic option, but they have traditionally been too computationally intensive for interactive applications. Recent advances make wave approaches viable, but these still require extensive precomputation and thus are unsuitable for a plug-in in which users may frequently change scene geometry [7]. This leaves geometric sound propagation algorithms, which approximate wave-propagation results by making the assuption that sound travels as rays, thereby enabling leveraging ray-tracing techniques from computer graphics.

GSound, a geometric sound propagation library from the GAMMA group at UNC-Chapel Hill, boasts the cutting edge of geometric sound propagation techniques, making it ideal for this application [8, 9, 10]. Originally developed for interactive gaming and VR applications, GSound was designed for accuracy and efficiency, particularly in dynamic scenes [8, 10]. This runtime performance is a prerequisite for integrating into the DAW workflow, where tens of plug-ins will be chained together, and an unresponsive plug-in will quickly be terminated and ignored. GSound also offers several advantages beyond speed:

- The quality falloff when reducing the number of rays is limited [8]. This allows audio engineers to temporarily reduce GSound's computational footprint if necessary to maintain realtime performance, without a marked decrease in sound quality.

- Very large scenes are supported, complete with specular and diffuse reflections, as well as high-order edge diffraction [9]. These amount to a more realistic rendered result.

- Head-related transfer functions (HRTFs) are supported. HRTFs encapsulate the effects of sound interacting with a listener's unique head and ears that are used for localization.

This combination of factors lead to selecting GSound to power this sound propagation plug-in, which we naturally call GSoundAU.

3.2 Algorithm Overview

Before diving into the design of GSoundAU, it is worthwhile to give a high-level overview of GSound's propagation techniques to better appreciate its advantages. GSound's core approach, noting that nearby geometry generally has the largest perceptual impact on perceived sound, is based on shooting visibility rays from the listener, in contrast to many other source-to-listener systems [8]. This approach means that the ray count is not proportional to the source count and fewer rays can be shot to increase efficiency. (This is a noteworthy feature for other potential GSound applications, but since DAW plug-ins only work with a single audio stream, this particular speed improvement is not leveraged.)

In the core algorithm, a random sphere of rays propagates from the listener and reflect off the scene geometry up to a maximum specular reflection depth. These sequences of reflections are tracked in hash tables for each reflection depth level, which prevents duplicate paths and enables reflection sequences to be tracked between frames [8]. Tracking reflections this way ultimately allows for both fewer rays per frame and improved audio consistency between frames. For each triangle in a visit sequence, a listener image is made. Then, for each valid path, audio is calculated, incorporating the material absorption and scattering coefficients and relative velocities of the source and listener [8]. A radiosity-based approach is used for the diffuse reflections, and edge diffraction is approximated using Kouyoumjian and Pathak' uniform theory of diffraction (UTD) [8, 9].

4 GSoundAU

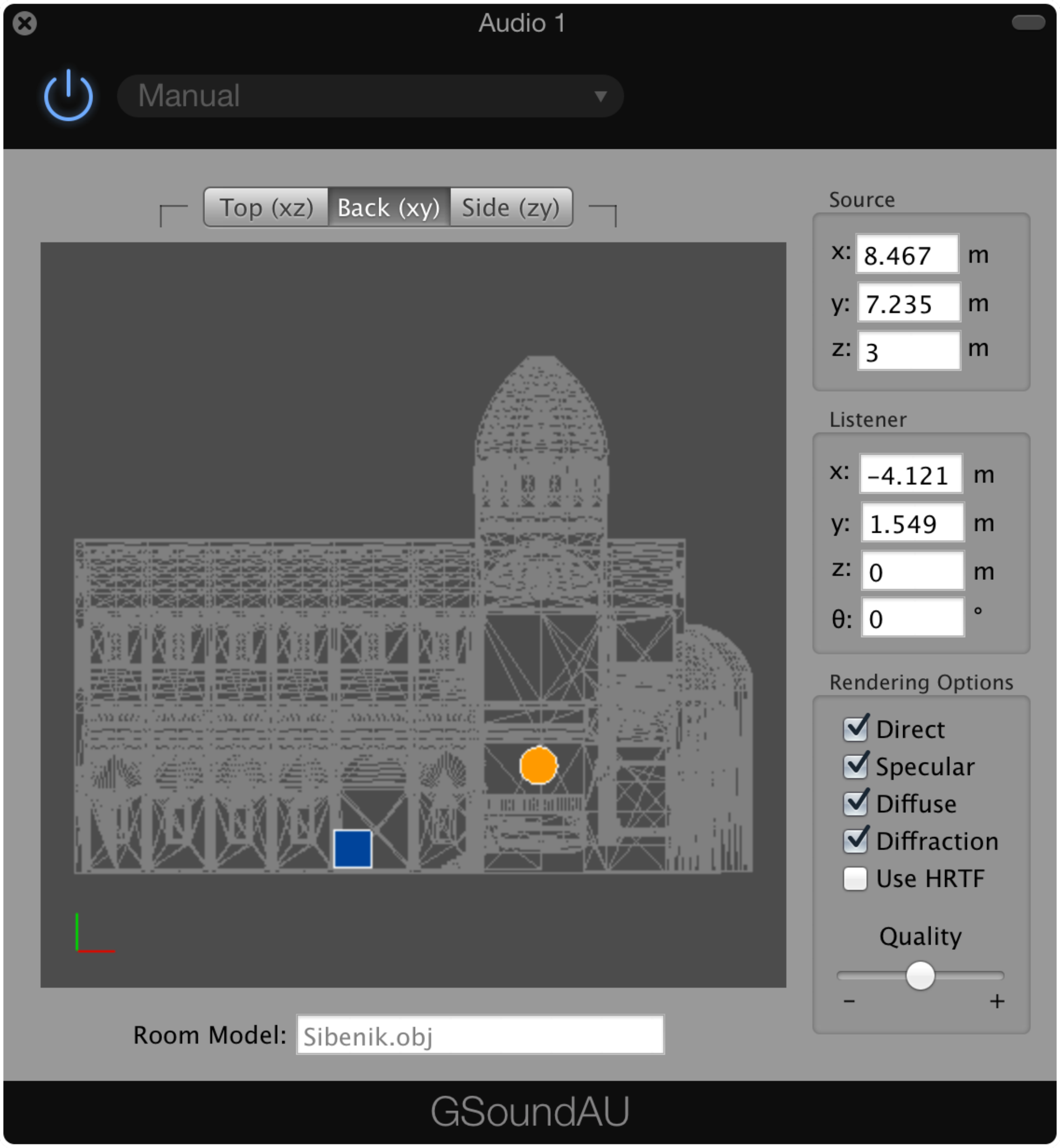

Figure 3: This is GSoundAU, as hosted by Apple's GarageBand. Dragging the blue arrow or editing the text boxes moves the listener position, and the source can be similarly manipulated. Right-clicking rotates the listener to face the clicked point. The Top (xz), Back (xy), and Side (zy) buttons switch the perspective on the model to allow for manipulating the positions in other planes.

4.1 User Interface Overview

GSoundAU integrates seamlessly into typical DAW workflows. Users simply select it from their list of available plugins and add it to a track as usual. This works as both a mono-to-stereo and a stereo-to-stereo plugin, meaning no extra steps need be taken to change a track's number of audio channels. Once the plugin is launched (Figure 3), users select an object file containing the model of the environment, which then gets displayed as a wireframe rendering and can be viewed from the three angles often used in 3D modeling software: the top, back, and side.

Users create the propagation effects they want by simply moving the positions of the source and the listener with a click and drag. This makes for intuitive reverb effects- instead of worrying about arcane convolution reverb parameters, audio engineers can now get the sounds they want by imagining themselves in a scene. If more precise control is needed, the exact numbers can be edited in a side panel, and animations can be achieved using the typical DAW graph interface, as shown in Figure 4, or using the DAW's plugin parameter recording tools. Also, if only some propagation effects are desired, another side panel allows for enabling or disabling several components of propagation, as well as the HRTF.

Figure 4: Full integration with DAW hosts allows users to animate GSoundAU parameters (such as the x coordinate of the source position) in the main timeline window.

4.2 User Interface Design

Being conceptually different than the reverb plug-ins that audio engineers are accustomed to, GSoundAU's interface design must quickly and effectively communicate how to use it. A clean, minimal design is used to that end, and native UI components further add a sense of familiarity. As seen in Figure 3, these components are placed in English left-to-right, top-to-bottom reading order, to present a use story for new users and to prioritize the critical elements. For instance, the model source file will typically be set only once for a given project, so it is on the bottom- though still clearly grouped with the projected display.

The centerpiece of the design is the view for placing the source and listener, since these are the primary parameters being manipulated. In fact, since the source and listener rest on the floor for the vast majority of use cases, engineers will spend most of their time making floor plane translations. This, coupled with the fact that human brains are better at pattern matching in 2D over 3D, motivates using a top-down 2D view as the default for source and listener placement [11].

Naturally there is still a need for 3D translations. As with any aspect of information visualization, it is important to consider what perceptual task the visualization should be optimized for. Smallman et al. found that "3-D views are useful for understanding object shape, but 2-D views are more useful for understanding the relative positions of objects"[12]. For GSoundAU, the primary task is positioning the source and listener, so, somewhat counterintuitively, 2D views are preferred.[4] Following in the tradition of commercial object modeling software, back and side views are offered, and, as an additional nod to users with that background, the red, green, and blue axes directions are discretely placed in the lower corner. Using wireframe renderings in all three views serves to give some sense of the shape without interfering with the primary positioning task.

Color choices were largely motivated by Colin Ware's guidelines [11]. Specifically, the glyphs representing the source and listener are outlined in white and much more saturated than the background and wireframes, so as to immediately grab the users' attention and be easily found. They also have enough yellow-blue variation to work for color-blind people.

[4] If, in future versions of the plug-in, an alternate 3d view is included, it should use linear perspective, since that has been demonstrated to be the most useful depth cue for translation tasks [11]. It should also feature cast shadows, the most effective cue for height [11].

Each element and its surrounding whitespace was carefully considered before being included and placed. Space comes at a premium for plugins, because users will often have multiple ones open at once and still want to see the DAW's timeline. This requires some sacrifices: though there would likely be some advantage in the positioning task from seeing all three projections at once, that would either make for an overlay-large plugin or too-small projection views. Similarly, several features were considered, but ultimately rejected. Some, like a gain parameter, can be produced by DAWs without a plugin.[5] Others, like custom HRTF loading, would only be useful to a very small slice of the target user-base, so GSoundAU just offers a generic HRTF.[6] The current GSoundAU screen footprint appears to be right in line with commercially available high-end plugins, while managing to maximize screen real estate for use in the primary manipulation tasks.

4.3 Development Issues

Once installed, GSound proved to be a rich tool for many sound propagation needs. It can handle many more sources and much more dynamic scenes than are required for this plugin, and is clearly ready-made for gaming and virtual/augmented reality applications. For use in this plugin, GSound was less plug-and-play, because the exposed methods for working with audio at the sample level are limited. It is worth noting that this is by design of course, since most audio applications work at the higher level of abstraction of loading and playing full audio clips.

The real challenges in GSoundAU's development came from Apple's limited and often broken support for Audio Unit development- particularly for Audio Units that aren't DSP-based or have custom views. The documentation is largely outdated and is mostly based around sample code that won't compile on many development platforms. Making a backwards-compatible audio unit proved sufficiently difficult that GSoundAU is only officially supported

[5] Plugins often seem to fall into this trap. RaySpace, for example, includes a four-band equalizer in the main view that is inherently less flexible than the equalizers built into every prominent DAW.

[6] Most audio engineers are producing audio for many people to consume; it's up to their discretion whether the generic HRTF will enhance localization for enough of their audience.

by newer DAWs (specifically DAWs that have fully dropped outdated Carbon graphics requirements).

5 Conclusion

5.1 Summary

GSoundAU serves audio engineers by integrating easy-to-use, fast, and accurate sound propagation effects into the standard DAW-based workflow. The plug-in can easily be used like a much more accurate static reverb plug-in, or users can animate a scene to place the listener and source in an environment- without worrying about manually mimicking occlusion. Outside of professional applications, by integrating with Apple's GarageBand, GSoundAU lowers the barrier to experimenting with sound propagation.

5.2 Limitations and Future Work

Still, more work remains to be done- starting with incorporating feedback from audio engineer alpha testers. Besides that, the largest outstanding challenge for improving GSoundAU is that of allowing users to specify the audio material properties of the scene geometry. This needs to be a rapid process to integrate smoothly into typical audio engineering workflows, but traditional 3D modeling software techniques like individual polygon selection or perspective painting require time and practiced 3D manipulation skills. Currently the most promising option is to allow users to select walls, floors, and ceilings by polygon angle to the ground plane to at least allow for setting material properties of the largest patches of the scene, but this is an area of ongoing consideration. Fortunately, GSound already supports loading and saving material properties, which will facilitate the eventual user experience solution.

6 Acknowledgements

Thanks to Dr. Manocha for advising this project, Carl Schissler for GSound and design advice, and Dr. Lin for reviewing this paper.

References

[1] Apple Inc. Logic Pro X: In Action. http://www.apple.com/logic-pro/in-action/.

[2] Inc. Wave Arts. Panorama. http://wavearts.com/products/plugins/panorama.

[3] Universal Audio. Ocean way studios plug-in. http://www.uaudio.com/store/reverbs/ocean-way-studios.html.

[4] SilverSpike. Roommachine 844. http://www.silverspike.com/?Products:RoomMachine_844.

[5] Waves. Nx virtual mix room over headphones. http://www.waves.com/plugins/nx.

[6] QuikQuak. Rayspace. http://quikquak.com/prod_rayspace.html.

[7] Ravish Mehra, Atul Rungta, Abhinav Golas, Ming C. Lin, Dinesh Manocha. WAVE: Interactive Wave-based Sound Propagation for Virtual Environments. *Proceedings of IEEE VR*, 2015.

[8] Dinesh Manocha Carl Schissler. GSound: Interactive Sound Propagation for Games. *Proc. of AES 41st Conference: Audio for Games*, 2011.

[9] Carl Schissler, Ravish Mehra, Dinesh Manocha. High-Order Diffraction and Diffuse Reflections for Interactive Sound Propagation in Large Environments. *Proceedings of ACM SIGGRAPH*, 2014.

[10] Carl Schissler, Dinesh Manocha. Adaptive Impulse Response Modeling for Interactive Sound Propagation. *Proceedings of i3D*, 2016.

[11] Colin Ware. *Information Visualization: Perception for Design, ch. 7: Depth Cues in Combination*. Elsevier, 3rd edition, 2013.

[12] Smallman, Harvey S and Cowen, Michael B et al. Human factors of 3-D perspective displays for command and control. Technical report, DTIC Document, 2002.