

## **AMD TrueAudio Next – The right next step for audio on VR**

*Carl Wakeland, Fellow Design Engineer, AMD*

Virtual reality demands a new way of thinking about audio processing. In the long history of games and video on flat screens, the standard of realism for audio rendering has remained relatively simple, especially when compared against contemporaneous advances in graphics and cinematic video rendering. Although hearing is inherently a 3-dimensional sense, audio for flat screen games and cinema/video has usually minimized the use of 3D and other advanced audio rendering technologies, for the simple reason that all of the graphics and video is in front of you. With a 2D screen in front of you, if you hear a sound of a voice or an object behind you and turn toward it, all you will see is perhaps a speaker or the walls of your apartment. With the exception of some FPS games where 3D audio provides a tactical assist that players learn to use to their advantage, too much realism in audio for a flat screen game or movie can sometimes be a distraction, especially if it is inconsistent with the visual experience. For example, surround sound in cinema almost universally uses rear and side speakers for ambient fill effects, and almost never for significant auditory cues, because to do so would distract the audience from the action on the screen in front of them.

A head-mounted display changes everything. Now the user can turn in any direction and see a continuous visual scene, and with recent advances, walk independently through a virtual world. Advanced virtual reality systems are expected to give the user a sense of presence approaching consensus reality – a “perceptual illusion of non-mediation”. Studies (Larsson et. al, 2010) have shown that realistic audio is a significant prerequisite for establishing presence in virtual reality.

What are the ingredients of realistic audio? It is commonly stated that accurate spatial and positional audio rendering using head-related transfer functions (HRTFs) is sufficient to create realistic audio. While this can be true for scenarios where the user is fixed in one position or placed on a “magic carpet ride”, and the audio designer includes all of the environmental effects of reverberation, occlusion, reflection, diffraction, absorption and diffusion in every pre-recorded sound, with HRTFs taking care of positioning each sound in the user’s ears, the process becomes inadequate as soon as the user is allowed to move freely in the scene, even if within a limited area. When the user is in motion or changing the center position of their head, reflection paths and environmental effects for every sound are changing continuously. It is no longer practical to pre-bake the environmental effects for every sound in the scene. The typical shortcut approximation taken is to lump all of these effects into a reverberation plugin, in some cases using one reverb setting for the entire scene, or having multiple settings distributed to different rooms in the scene. In the latter case, a sound originating from an adjacent room may be processed with a different reverb setting than a sound originating from the same room as the listener. Technologies that provide these rendering capabilities have been in use since the 1990’s.

For the listener in a virtual world, these approximations are not able to create presence, even if the sounds are very accurately positioned. Consider an example where a VR user is walking down a hallway, hears a sound in front of them emerging from a room with an open door into the hallway, and walks past the doorway while the sound continues to play. In the physical world, the user would be hearing continuous changes in the sound’s environmental effects:

- Occlusion of the room walls,
- Diffraction around the doorway,
- Reflections off of the surfaces of the walls, floor, and ceiling,
- Diffusion and absorption from the materials making up the surfaces of the interior walls/floor/ceiling of the building, and its objects/furniture.

The conventional audio design and rendering approach of abstracting a room reverb in a studio, adding simple attenuation and low-pass filtering of a sound source, and positioning it with an HRTF, creates a credible representation of a sound, but fails to create presence. This is true even when the audio designer takes pains to use realistic curves for distance attenuation and filtering of sound sources, and continuous updates to the HRTF positions as the listeners' ears or the source change position. The reason that conventional approaches fall short is that real-world acoustics are considerably more complex than the scope of these approximations, and human brains are well-trained by exposure and adaptation to recognize the acoustics of the real world and to discriminate them precisely. Human hearing is a critical survival adaptation -- considering that sound is often a first indicator of danger, and the ability to assess the direction and distance of a given sound while picking it out from a noisy environment is a critical adaptation skill. A present-day example of this capability in action is the so-called "cocktail party effect".

Bringing environmental sound rendering closer to real-world acoustics requires modeling of the physics that propagate sound; this is called auralization. There are multiple approaches that have been proposed and implemented for acoustic propagation modeling that make varying tradeoffs between complexity and accuracy. Perfect modeling -- e.g., solving the acoustic wave equation for every sound propagation event -- is still not within practical reach of current real-time compute capabilities for VR systems, but with the power of real-time GPU compute enabled by AMD TrueAudio Next, significant improvements to auralization can be made that are not practical to achieve on CPU alone. One approach that provides a significant upgrade in realistic auralization for audio occlusion and reflections in critical frequency bands is the method of geometric acoustics.

Geometric acoustics starts with ray tracing of paths (typically a sampled subset) between each sound source and the position of the listener's ears, and applies a collection of algorithms (e.g., acoustic radiance transfer) to the dataset of material properties encountered in the path bounces, to generate a unique impulse response for each sound, per ear. In addition to path reflection, diffusion and occlusion, diffraction effects (e.g., finite edge diffraction) and HRTF filters can also be modelled within this framework and superimposed into each of these time-varying impulse responses (Chandak, 2011).

In the realtime rendering process, the impulse responses, which are continuously updated as the sources and listener change position, are convolved with the corresponding audio source signals. These signals are then mixed separately per ear to generate the output audio waveforms heard by the listener. The approach is scalable and has been implemented both on CPU and with AMD TrueAudio Next. The number of physically modelled sound sources that can be supported is significantly enhanced with TrueAudio Next; instead of being limited to a small handful of primary sound cues, an application can scale so that environmental sound sources (up to a complete soundscape of 40 to 64 sounds) can be included, by "borrowing" a small (roughly 10-15%) subset of a GPU's compute units. Quality can be scaled in even greater dimensions when multiple GPUs or combinations of APUs and GPUs are deployed.

Two primary algorithms required for geometric acoustics rendering are time-varying convolution (in the audio processing component) and ray-tracing (in the propagation component). On AMD GPUs, ray-

tracing can be accelerated using AMD's open-source FireRays library, and time-varying real-time convolution can be accelerated with the AMD TrueAudio Next library.

The AMD TrueAudio Next library is a high-performance, OpenCL-based real-time math acceleration library for audio, with special emphasis on GPU compute acceleration. In addition to low-latency, time-varying convolution, the TrueAudio Next library also supports highly efficient FFT and Fast Hartley Transforms (FHT) and complex multiply. More is on the way; the AMD TrueAudio Next library is being released to open-source under GPUOpen.

The TrueAudio Next library can run on x86 CPU or AMD GPUs that support GFX8. An SDK with sample audio rendering examples is included.

• • •

Having established the use case above with the TrueAudio Next library and FireRays SDK as key solutions, two critical questions remain:

- 1) Can real-time audio rendering technology be supported on the GPU compute shaders without interfering with graphics rendering and causing judder and/or critical frame rate loss?
- 2) Can high-performance GPU audio really be rendered glitch-free, with low latency, in a VR gaming or advanced cinematic rendering scenario?

Although conventional wisdom has told us that audio rendering on the GPU causes unacceptable latency and interferes with graphics performance, the answer to both questions is yes, and leads to the other main pillar of AMD TrueAudio Next: Compute Unit (CU) Reservation with asynchronous compute.

AMD's asynchronous compute technology is already well-known in the VR rendering space as a key component of Liquid VR's Time Warp and Direct-to-GPU rendering features. Instead of making all graphics shader functions wait in a single queue to execute on the entire array of CUs, asynchronous compute allows multiple queues of functions to use different sets of CUs simultaneously, with variable execution priorities, under control of an efficient hardware scheduler.

AMD's CU Reservation feature takes this idea a step further: a limited set of CUs can be partitioned off and reserved for as long as is required for an enabled application, and accessed through a reserved real-time queue. For example, in a GPU having 32 CUs, 4 or 8 might be reserved for exclusive use by TrueAudio Next, with the remaining 24 to 28 fully available to graphics. The reservation is performed entirely within a TrueAudio Next enabled application, plugin or engine; not at boot time, and the CUs are freed when the application releases them or exits. Moreover, an additional "medium-priority" queue may be assigned to the reserved CUs for slightly lower priority kernels. In the case of time-varying convolution, the audio data path, which must be low-latency and absolutely glitch-free, may use the real-time queue, while the slightly less critical impulse response update uses the medium-priority queue.

CU Reservation offers a number of critical benefits that enable audio to co-exist with graphics:

1) The number of CUs that are reserved (if any) is flexible, scalable and totally at the discretion of the game developer, under guidance of the plugin and audio engine provider's recommendations. Audio engines have long experience with scaling to available CPU resources using excellent profiling tools. AMD TrueAudio Next simply adds a higher dimension –a big, reliable, configurable, private sandbox.

2) Surprises are avoided. CU Reservation allocations can be established early in a game's development cycle. Audio and graphics design can proceed independently without fear that audio can inadvertently "steal" any more graphics compute resources than were allocated to it at the start of development. CU Reservation actually provides a tighter (but much larger) sandbox than multi-core CPUs running a general-purpose OS can possibly provide.

3) Graphics is isolated from audio, and audio is isolated from graphics. Only memory bandwidth is shared, of which audio takes a very small share when compared with graphics, and for which DMA transfer latencies are far more than adequate. Glitch-free convolution filter latency as low as 1.33 ms (64 samples @ 48 kHz) has been achieved with over 2 second impulse responses. Typical audio game engines require 5 to 21 ms total buffer latency.

CU Reservation is a driver feature that is provided to NDA partners. The TrueAudio Next library can be used with or without CU Reservation.

The AMD TrueAudio Next open-source library and driver-controlled CU Reservation will enable dramatically higher levels of audio rendering realism in VR. We can't wait to hear what you will create with it!