Real-time 3D Audio Downmixing System based on Sound Rendering for the Immersive Sound of Mobile Virtual Reality Applications

Dukki Hong[1], Hyuck-Joo Kwon[2], Cheong Ghil Kim[3], and Woo-Chan Park[1*]
[1] Department of Computer Engineering, Sejong University, South Korea
[e-mail: dkhong@rayman.sejong.ac.kr, pwchan@sejong.ac.kr]
[2] School of Electrical & Electronic Engineering, Yonsei University, South Korea
[e-mail: hyuckjookwon@yonsei.ac.kr]
[3]Department of Computer Science, Namseoul University, South Korea
[e-mail: cgkim@nsu.ac.kr]
*Corresponding author: Woo-Chan Park

Received September 27, 2017; revised February 14, 2018; revised April 15, 2018; accepted May 5, 2018; published December 31, 2018

Abstract

Eight out of the top ten the largest technology companies in the world are involved in some way with the coming mobile VR revolution since Facebook acquired Oculus. This trend has allowed the technology related with mobile VR to achieve remarkable growth in both academic and industry. Therefore, the importance of reproducing the acoustic expression for users to experience more realistic is increasing because auditory cues can enhance the perception of the complicated surrounding environment without the visual system in VR. This paper presents a audio downmixing system for auralization based on hardware, a stage of sound rendering pipelines that can reproduce realiy-like sound but requires high computation costs. The proposed system is verified through an FPGA platform with the special focus on hardware architectural designs for low power and real-time.

The results show that the proposed system on an FPGA can downmix maximum 5 sources in real-time rate (52 FPS), with 382 mW low power consumptions. Furthermore, the generated 3D sound with the proposed system was verified with satisfactory results of sound quality via the user evaluation.

Keywords: 3D Audio, audio mixing system, virtual reality, sound rendering, ray tracing, FPGA

http://doi.org/10.3837/tiis.2018.12.018 **ISSN : 1976-7277**

1. Introduction

Recently, the interest in virtual reality (VR) has increased with the help of the advanced developments in graphics, mobile application processsors (APs), and sensory I/O technology, rapidly. A good case is that Facebook took over Oculus, a famous for VR company in 2014; commonly, it is understood that VR technology will be an important trend in the future. Also, experts of Wall Street predicted that the future of VR will be as one of the most significant technologies ever [33]. The realistic acoustic spaciousness should be reproduced to create an immersive VR experience in mobile devices because auditory cues can enhance the perception about the complex surrounding environment without the visual system [1]. Therefore, the need for 3D sound technology is increasing for product differentiation and competitive strategy in consumer automotive electronics fields such as tablets, smart phones, sound bars, TVs, and set-top boxes.

Generally, two technologies have been used to create enveloping 3D sound with acoustic spaciousness: multi-channel audio systems and Head Related Transfer Function (HRTF) [2, 3]. However, it is understood that both have limitations to reproduce realistic 3D sounds; the former needs as many as speakers with their installation space to achieve more 3D sound quality; the latter may not produce full realistic 3D sound because it is not able to reflect the physical effects on the surrounding environment and the surface material of the geometry model perfectly in the virtual space. To solve these problems, leading companies and universities have announced 3D sound technologies based on sound rendering [4, 5, 6], which can reproduce more sensible auditory spatial feeling through physical simulation. For this purpose, sound rendering process could be divided into three stages; the first one is the sound synthesis stage to create a dry sound; next is the sound propagation stage to simulate the physical phenomenon of sound; the last one is the sound generation stage to create 3D sounds (wet sounds) using simulation results and dry sounds.

In mobile systems, there must definitely be many difficulties with generating high quality multimedia contents including sound rendering due to power consumption and heavy computational complexity. Especially, in terms of real-time processing, sound rendering has the hurdle that only high performance PC can allow real-time processing. This issue gets more worse on mobile platforms because mobile APs have power consumption issue. Furthermore, interactive sound rendering applications such as VR require that the processing performance of the sound generation stage should be 20 to 100 FPS; it means that high quality sound can be provided to the user with applications having a large number of sound sources [7]. If not guarantee the performance condition, inaccurate wet sounds can be produced. As a result, the listener may hear low-quality 3D sound with unpleasant noise.

There are generally considerate methods to accelerate the sound generation stage for real-time auralization. One approach is to accelerate by using mobile general purpose (GP) multicore or manycore processor. In mobile platforms, this approach is not suitable because of heavy power consumptoins by high processor utilizations for acceleration. The other is to design a dedicated hardware. According to [8], a dedicated hardware can provide greater energy efficiency up to 50-500 times than GP processors in some cases. For this reason, we chose the second approach to accelerate the sound generation stage because it can be more effective for mobile platforms.

In this paper, a 3D audio downmixing system based on sound rendering for immersive VR applications sound is proposed. To reduce power consumption, which is a major issue of the mobile platform, sound generation stage of the sound rendering pipeline is designed as a dedicated hardware. By implementing with dedicated hardware, the proposed system can mitigate power consumption problems as well as achieve real-time performance. Furthermore, it is possible to reproduce sound effects such as absorption, diffraction and reflection that conventional HRTF solutions and multi-channel audio can not express properly.

The proposed 3D audio downmixing system was implemented using Xilinx Artix-7 XC7A200T. The memory/logic utilizations of the proposed system was measured as 56% and 33%, respectively, for the performance evaluation. We measured the power consumption with the Xilinx Power Estimator (XPE). In the normal case, the result was 382 mW. Also the qualitative assessment with quality test was conducted to measure the user sound experience. The result showed the satisfactory quality compared with commercial downmixing product.

This paper is composed as follows. In Section 2, related work is described. In Section 3, the proposed system is described in detail. In Section 4, we present the hardware implementation on an FPGA. Section 5 presents the experimental results and its analysis. Finally, the conclusions and future work are presented in Section 6.

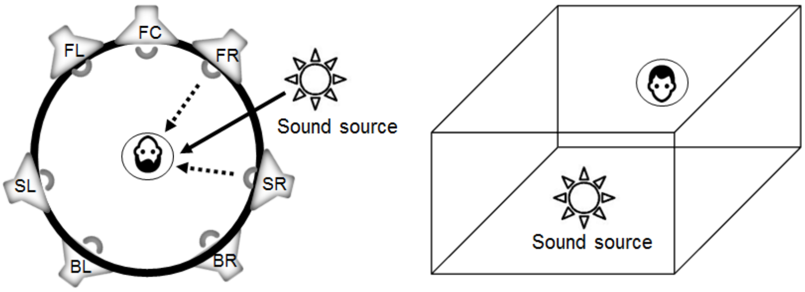
2. Related Work

In general, acoustic spaciousness can be reproduced using either 3D sound technology based on multi-channel audio or HRTF. A multi-channel audio system is a technology that provides surround sound in all directions by arranging several speakers on the front, rear, left and right sides [9, 10], such as Dolby ATMOS and Auro-3D. HRTF is a method of calculating the difference between the intensity and delay time of sound reaching both ears from a specific sound source. The technologis based on this are AM3D's ZIRENE 3D, OpenSL ES 3D Audio, and Oculus Audio SDK [11, 12, 13]. Both methods, however, have some limitations to reproduce realistic acoustic spaciousness.

The multi-channel audio system requires a dedicated loudspeaker system with installation space for them. Moreover, additional operations are required if the sound source in virtual space is located between speakers placed around the listener. For example, looking at **Fig. 1** (a), the source is between the speakers FR and SR. In this case, it is difficult to map the sound source directly to a specific speaker. For this reason, the audio system should perform additional operations so that the corresponding sound source can be output to the speaker FR and SR respectively.

Most HRTF-based 3D sound technologies use simple scene models such as the shoebox model shown in **Fig. 1 (b)** [13]. There is a limit to reproduce realistic sound because the characteristics of the surrounding environment and the surface material of the geometry model on the virtual reality are not reflected.

Sound rendering is regarded as a techniques to overcome these limitations by simulating and tracing the sound propagation paths between the listener and sound source. Sound rendering can be classified into two categories: the numerical method and the geometric method [14].



(a) (b)

Fig. 1. (a) An example of limitations of multi-channel audio systems. **(b)** An example of a shoebox model.

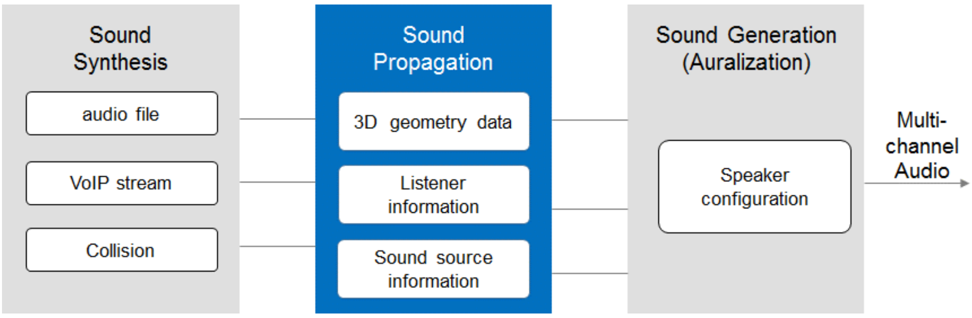


Fig. 2. An example of sound rendering pipeline

The numerical method solves the wave equation mathematically to process sound propagation [15, 16, 17]. The result of the numerical method is more accurate than the geometric method, but an interactive rate processing could not be achieved because of a huge amount of calculation cost [14]. For this reason, the numerical method has been used only for processing static scenes.

The geometric method, in which a sound wave is as treated as a single ray, is used widely to process sound propagation at an interactive rate. The geometric method is a technique to calculate the characteristic value of sound by tracing the sound propagation paths between the listener and the sound source [18]. Sound propagation path means a valid path where sound originating from a sound source position reaches the listener (see Fig. 3).

So for many researchers on the geometry method have suggested ways to find more sound propagation paths. A source clustering by grouping visible neighbors and a expanding the existing image source mehtod to a spherical sound source were proposed [19]. It achieved plausible acoustic effects at interactiverates in large dynamic environments containing many sound sources. Interactive sound propagation technology [20] was introduced using bidirectional sound transport based on bidirectional path tracing and it found more propagation paths than conventional bidirectional path tracing.

Also world-renowned companies have introduced sound rendering technology based on the geometric method. Nvidia has announced VRWorks Audio based on its path-tracing technology, OptiX [21]. AMD announced TrueAudio Next based on OpenCL and path tracing, and is releasing its software development kit (SDK) [22]. Valve has announced Steam Audio,

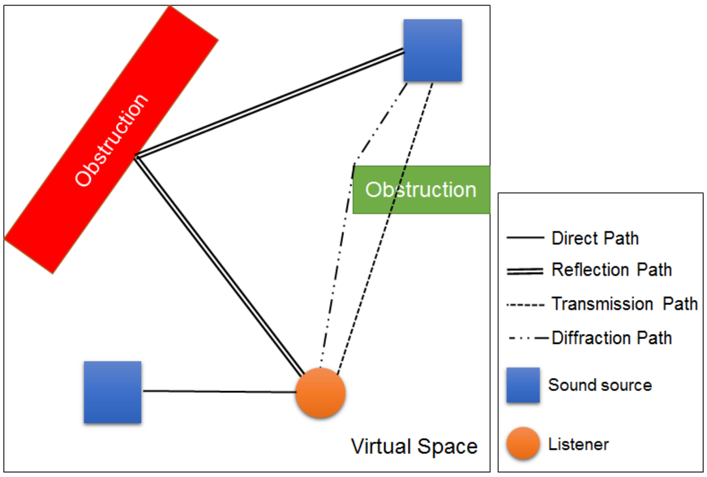


Fig. 3. Examples of sound propagation paths. The direct path is a sound path between the listener and the sound source, with no obstructions. Reflection and diffraction path is a sound path through which the sound reaches the listener after it collides with the obstacle, and the transmission path is a sound path through which the sound is transmitted to the listener when there is an obstacle between the listener and the sound source.

immersive audio solutoins for games and VR, by acquiring Impulsonic which is a 3D sound technology company [23].

The sound rendering pipeline consists of the sound synthesis, the sound propagation, and the sound generation (or auralization) in **Fig. 2**. During the process of sound rendering, both stages of sound propagation and the sound generation are the most important stages to give immersive feeling with the cost of high computatinal complexity and long processing time. Acceleration of these stages also influences real-time processing of sound rendering.

Sound synthesis, the first stage of the sound rendering pipeline, creates a sound effect based on an event of user interaction. That is, it processes sounds that occur when a user drops something or knocks on a door, and is similar from a technique generally used in existing UIs and games. The sound propagation stage simulates the propagation between the sound source and listener on VR environments, as shown in **Fig. 3**. This stage calculates the sound characteristics (diffraction characteristics, reflection coefficient, transmission coefficient, absorption coefficient, and etc.) and the acoustic environment based on the geometrical characteristics of VR environments. The sound generation stage is to auralize wet sounds based on the configuration from the listener’s headphone or speaker using the characteristic values of sound calculated in the sound propagation stage.

3. The Proposed 3D Audio Downmixing System

In this section, we introduce the proposed 3D audio downmixing system, a dedicated hardware to accelerate the sound generation stage of geometric sound rendering [24], to reproduce the immersive sound of mobile VR applications at real-time rate with low power consumption. We introduce the proposed system in section 3.1. Each detailed architectures of the proposed system is described in sections 3.2-3.5.

3.1 The Overall System Architecture

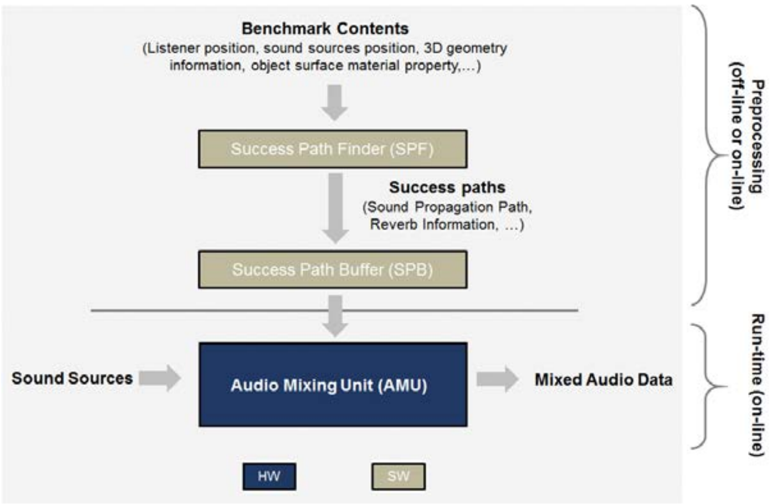

Benchmark Contents
(Listener position, sound sources position, 3D geometry information, object surface material property,...)
Success Path Finder (SPF)
Success paths
(Sound Propagation Path, Reverb Information, ...)
Success Path Buffer (SPB)
Sound Sources
Audio Mixing Unit (AMU)
Mixed Audio Data
HW
SW
Preprocessing (off-line or on-line)
Run-time (on-line)

Fig. 4. The architecture of the proposed audio downmixing system

Fig. 4 shows the overall architecture and processing flow of the proposed system. This system consists of the Success Path Finder (SPF) which finds a valid sound propagation path, the Success Path Buffer (SPB) which stores the information of found sound propagation path, and the Audio Mixing Unit (AMU) which generates wet sounds using the stored sound propagation path information. In SW, SPB and SPF are processed off-line or on-line according to application characteristics. In HW, AMU is processed on-line.

The proposed system's processing flow is as follows. Firstly, the SPF finds valid sound propagation paths by simulating the physical sound characteristics between sound sources and listeners in virtual space based on 3D geometry model. Secondly, the sound propagation paths found are stored to the SPB. For multiple 3D geometry data, this process is repeated. The AMU performs audio downmixing using dry sounds and sound propagation path data in run-time, and generates wet sounds for output of headphones and speakers.

3.2 The Proposed Audio Mixing Unit Architecture

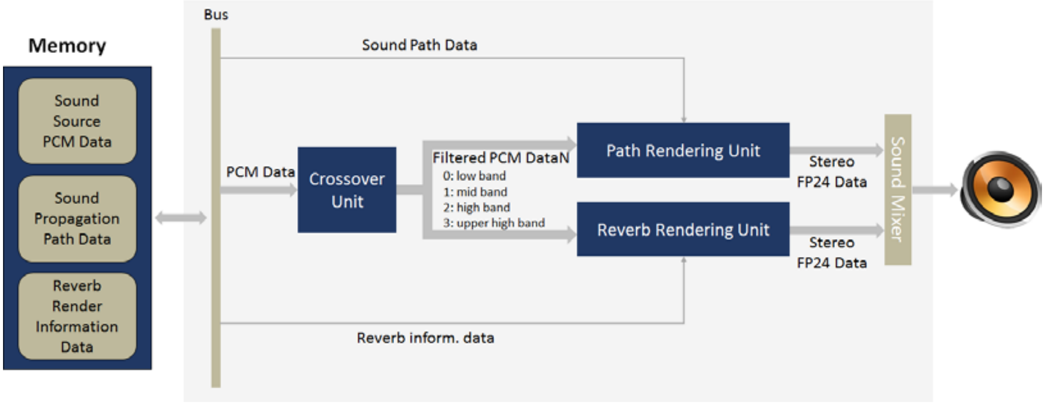


Fig. 5. The architecture of the proposed audio mixing unit hardware

The AMU downmixes n-channel (or n sound sources) to 2-channel. The AMU is designed to downmix a maximum 5-channel per single core in our architecture. We chose a fully fixed pipeline manner which has highly performance per energy for mobile platforms as mentioned in Introduction section.

The architecture of the proposed AMU is shown in **Fig. 5**. The datapath part consists of the Crossover Unit (CU) that separates PCM data by frequency band, the PRU that processes diffraction sound, reflection sound, transmission sound, and direct sound, and the RRU that processes reverberation sound. External memory stores data of reverb render information data, sound propagation path data, and sound source PCM data.

The CU is used to make audio processing more efficient according to frequency characteristics in the audio field, generally. It separates the signal into a predefined frequency band. Using sound propagation path information such as reflection and direct sound, the PRU reproduces an early reflection sound. The RRU reproduces the late reverberation sound that is expected as an important factor in feeling the ambiance such as the height of a hall and the area of a room. The information used in PRU and RRU are calculated in the SPF.

The AMU is processed as follows. Firstly, read dry sounds from the external memory for a predefined processing unit. According to the frequency band range, the CU separates the read dry sounds and sends it to the PRU and the RRU. The PRU uses the filtered data from the CU and the sound propagation path data found by the SPF to generate sound data with physical sound characteristics. At the same time, the RRU also generates sound data with sound attenuation using the filtered data from the CU and the reverb information data found by the SPF. The Sound Mixer mixes the sound data generated by the PRU and the RRU to produce wet sounds for output of headphones and speakers.

3.3 Crossover Unit

The proposed CU performs filtering on PCM data of dry sounds read from external memory according to the frequency band. We designed the CU in 4-way (upper high band, high, mid, and low), which is a commonly used approach. We selected the Butterworth high pass and low pass filters, which can allow flat passband response maximally to obtain a defined frequency band [25, 26].

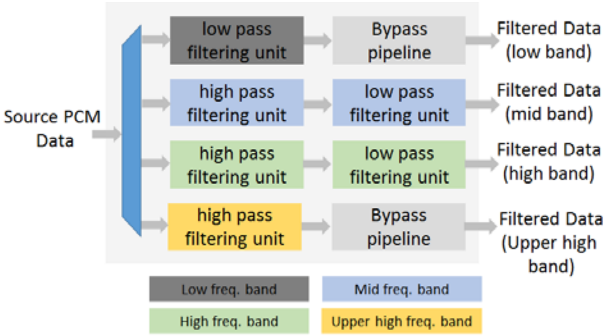


Fig. 6. The architecture of the proposed Crossover Unit

Fig. 6 shows the hardware architecture of the CU. The CU performs filtering on the PCM data transmitted through the bus according to the four frequency band regions. Low band is extracted from PCM data using low pass filter. Mid and high band are high pass filter, extract signal data after minimum frequency of corresponding band range, and filter by low pass filter to extract signal data of corresponding band range. Finally, to extract signal data after the high band, the upper high band performs a high pass filtering.

The upper high and low band filter units add a bypass pipeline to match the output timing with the high and mid band filtering units. The filtered signal data is transferred to the Reverb Rendering Unit (RRU) and the Path Rendering Unit (PRU) in the CU.

3.4 Path Rendering Unit

In Fig. 7, the PRU generates early reflection sounds by convolving the impulse response (IR) of the sound propagation path found in the SPF and the filtered signal data delivered from the CU. This unit consists of Propagation Path Renderer (PPR), Left/ Right PRU Delay Buffer, and Mixer. PRU Delay Buffer is a buffer to support delay sound effect and supports up to 25 ms delay time.

The PRU is processed as follows. First, from the CU, the 4-filtered signal data are transmitted to each PPR. At the same time, the IR data read from the external memory is transmitted to each PPR. Each PPR performs a convolution operation of IR and filtered signal data. When all PPR operations are completed, the Mixer mixes all rendered sound data for each channel and delivers the early reflection sound to the Sound Mixer.

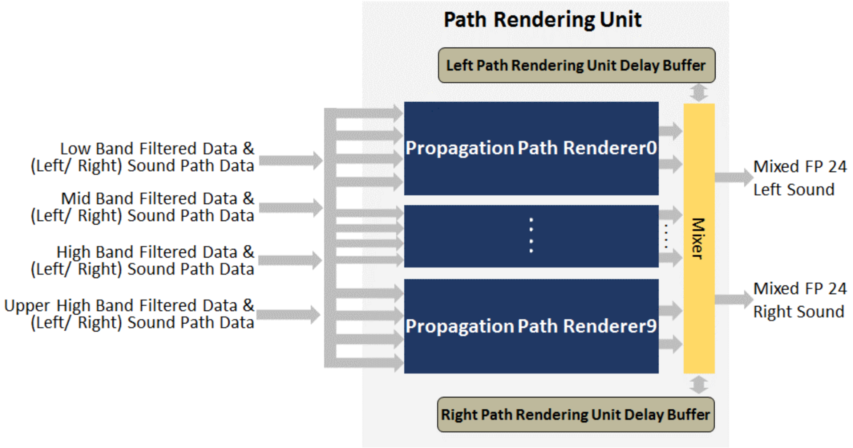

Path Rendering Unit
Left Path Rendering Unit Delay Buffer
Low Band Filtered Data &
(Left/ Right) Sound Path Data
Mid Band Filtered Data &
(Left/ Right) Sound Path Data
High Band Filtered Data &
(Left/ Right) Sound Path Data
Upper High Band Filtered Data &
(Left/ Right) Sound Path Data
Propagation Path Renderer0
⋮
Propagation Path Renderer9
Mixer
Mixed FP 24
Left Sound
Mixed FP 24
Right Sound
Right Path Rendering Unit Delay Buffer

Fig. 7. The architecture of the proposed Path Rendering Unit

A. Propagation Path Renderer

PPR (see Fig. 8) generates the rendered sound data for the corresponding sound propagation path through the convolution operation of IR and filtered data. This unit consists of two Channel Renderers which is consists of one Mixer and four Band Renderers. Each Band Renderer generates stereo sound data using the amplitude and delay information of the left/ right channel of the corresponding band range in the input IR. Our PPR allows concurrently to process up to 10 convolution operations using 10 PPRs. To simplify the convolution operation, we adopt a uniform partitioned frequency-domain delay line convolution [27]. Each delay line convolution is calculated using the following formula.

$$y[n] = g_{FB} y[n-N] + x[n] + (g_{FF} - g_{FB}) x[n-N] \tag{1}$$

Where n is a sample number, N is a delay in samples, g_{FF} is the gain, and g_{FB} is the feedback gain.

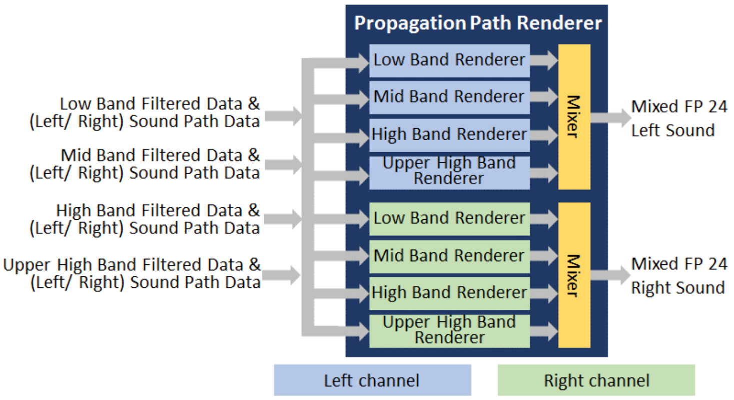

Propagation Path Renderer
Low Band Filtered Data & (Left/ Right) Sound Path Data
Mid Band Filtered Data & (Left/ Right) Sound Path Data
High Band Filtered Data & (Left/ Right) Sound Path Data
Upper High Band Filtered Data & (Left/ Right) Sound Path Data
Low Band Renderer
Mid Band Renderer
High Band Renderer
Upper High Band Renderer
Low Band Renderer
Mid Band Renderer
High Band Renderer
Upper High Band Renderer
Mixer
Mixer
Mixed FP 24 Left Sound
Mixed FP 24 Right Sound
Left channel
Right channel

Fig. 8. The architecture of the Propagation Path Renderer

3.5 Reverb Rendering Unit

Using the calculated reverb time and gain in the SPF, the proposed RRU generates late reverberation sounds. We chose a widely used model of the Schroeder's reverberator [28] to generate a realistic late reverberation sound. Here normal configuration uses 2-allpass filters and 4-comb filters, but 10 comb filters are used in the proposed RRU. As more comb filters are added, it is possible to generate a realistic reverberation sound. The allpass filter uses the same number as before.

The RRU consists of the Comb Filter Mixer and the Allpass Filter Renderer{0, 1} (see **Fig. 9**). The Comb Filter Mixer consists of 10 comb filters, which operate in parallel. The Allpass Filter Renderer{0, 1} implement the allpass filter part used in Schroeder's reverberator model.

The RRU is processed as follows. From the CU, the four filtered signal data are transferred to the Comb Filter Mixer. At the same time, the Comb Filter Mixer receives the reverb information read from the external memory. And then the Comb Filter Mixer performs comb filter using filtered signal data and reverb information. When the operation of the Comb Filter Mixer is completed, the Allpass Filter Renderer{0, 1} are sequentially performed to generate a late reverberation sound. For mixing with early reflection sound, the Sound Mixer receives the generated late reverberation sound.

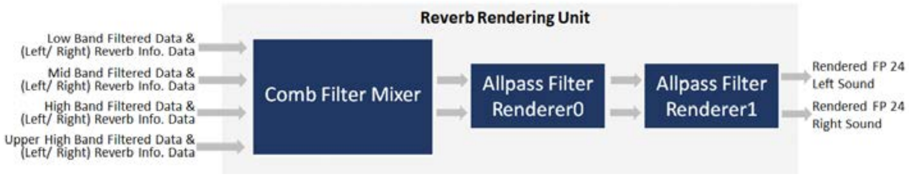

Reverb Rendering Unit
Low Band Filtered Data & (Left/ Right) Reverb Info. Data
Mid Band Filtered Data & (Left/ Right) Reverb Info. Data
High Band Filtered Data & (Left/ Right) Reverb Info. Data
Upper High Band Filtered Data & (Left/ Right) Reverb Info. Data
Comb Filter Mixer
Allpass Filter Renderer0
Allpass Filter Renderer1
Rendered FP 24 Left Sound
Rendered FP 24 Right Sound

Fig. 9. The architecture of the proposed Reverb Rendering Unit

A. Comb Filter Mixer

The architecture of the proposed Comb Filter Mixer is shown in **Fig. 10**. We design The Comb Filter Mixer with 10 comb filter renderers to generate realistic reverberation sound. The Comb Filter Renderer is designed to perform comb filter at every 4 bands simultaneously. The comb filter used in the Comb Filter Renderer includes a feedback comb filter, and this filter is calculated using the following formula.

$$y[n] = x[n-d] + gy[n-d] \tag{2}$$

where g is the feedback gain, d is a delay in samples, and n is a sample number. A delay buffer is placed for each comb filter renderer due to each comb filter renderer uses delayed output.

The operation of Comb Filter Mixer are as follows. When reverb information and four filtered signal data arrive at the Comb Filter Mixer, the Comb Filter Mixer delivers the filtered signal data to all comb filter renderers, and each renderer receives the reverb information according to the Comb Filter Renderer ID.

Each comb filter renderer performs a comb filter using the received data. When operations of all Comb Filter Renderer are complete, the comb filtered signals are transmitted to the Mixer. And the mixer mixes them into single sound data. Mixed sound data is passed to the Allpass Filter Renderer.

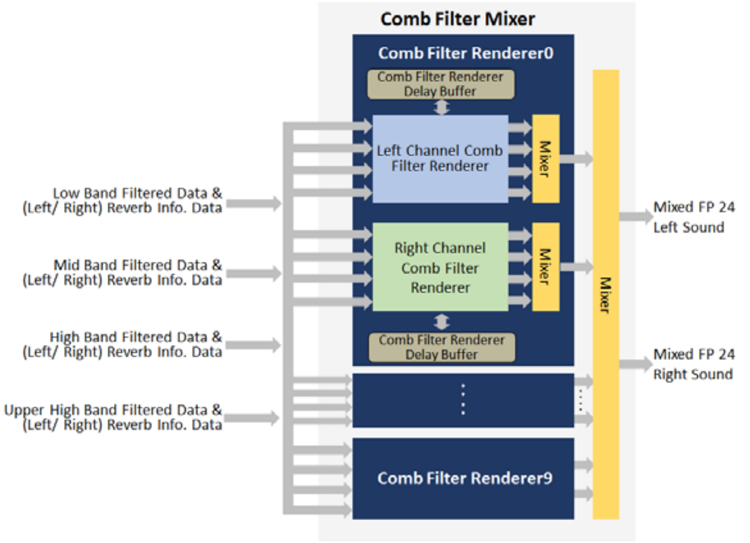

Comb Filter Mixer
Comb Filter Renderer0
Comb Filter Renderer Delay Buffer
Left Channel Comb Filter Renderer
Mixer
Right Channel Comb Filter Renderer
Mixer
Comb Filter Renderer Delay Buffer
Comb Filter Renderer9
Mixer
Low Band Filtered Data & (Left/ Right) Reverb Info. Data
Mid Band Filtered Data & (Left/ Right) Reverb Info. Data
High Band Filtered Data & (Left/ Right) Reverb Info. Data
Upper High Band Filtered Data & (Left/ Right) Reverb Info. Data
Mixed FP 24 Left Sound
Mixed FP 24 Right Sound

Fig. 10. The architecture of the proposed Comb Filter Mixer

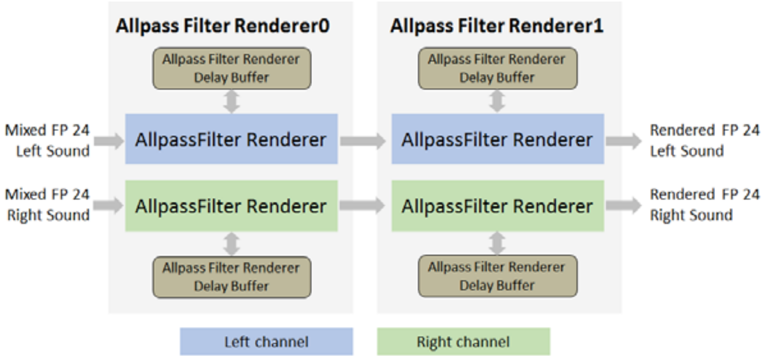


Fig. 11. The architecture of the Allpass Filter Renderer

B. Allpass Filter Renderer

The architecture of the proposed Allpass Filter Renderer is shown in **Fig. 11**. The Allpass Filter Renderer plays a role in contribution to intensity of reverb by using mixed sound data of the allpass filter and Comb Filter Renderer. Allpass Filter Renderer is comprised of the Channel Allpass Filter Renderer that performs allpass filter for each channel. In general, the allpass filter is designed by combining feedback comb filter and feedforward comb filter, and the following equation is calculated.

$$y[n] = -gx[n] + x[n-d] + gy[n-d] \tag{3}$$

where g is the feedback gain, d is a delay in samples, and n is a sample number. Since the Channel Allpass Filter Renderer uses delayed output like the Comb Filter Renderer, we assigned a delay buffer for each Channel Allpass Filter Renderer.

The Allpass Filter Renderer operates in the following order. First, the Allpass Filter Renderer0 performs allpass filter using feedback gain of the Comb Filter Mixer and mixed sound data and delivers the rendered sound data to the Allpass Filter Renderer1. As before , the Allpass Filter Renderer1 performs the same allpass filter and creates a late reverberation sound. The generated sound is passed to the Sound Mixer to produce the final sound data.

4. Hardware Implementation

In this section, the FPGA implementation of the AMU is described. The Dynalith Systems iMPRESS board with a Xilinx Artix7 XC7A200T chip and 1GB DDR3 DRAM is shown in **Fig. 12**. This FPGA board can be controlled from the host computer through USB 3.0 interface. **Table 1** shows the details of the Xilinx Artix7 XC7A200T FPGA chip, specifically.

Fig. 12. An iMPRESS FPGA board

Table 1. Xilinx Artix7 XC7A200T FPGA chip specification [29]

LUT	Logic Cell	BRAM Blocks	CMTs	Flip Flops
134,600	215,360	13,140 kbits	10	269,200
DSP Perf.	DSP Slices	GTP 6.6Gb/s Transceivers		I/O Pins
929 GMAC/s	740	16		500

As shown in **Fig. 13**, we implemented the Bus Functional Model (BFM), the AMU, the AMBA 64-bit AXI bus, the AXI master interface and the memory read/write controller on the Xilinx Artix7 XC7A200T chip. The external memory of the FPGA board consists of audio samples, mixed audio samples, reverb information, and propagation path information. Our FPGA prototype operates at a 50 MHz clock frequency.

A list of hardware resources for each unit is shown in **Table 2**. We use a 24-bit floating-point format (1 sign bit, 7 exponent bits, and 16 fraction bits) to reduce register requirements. In the AMU, the required SRAM size is 806.28 kbytes. The delay buffer sizes used in Comb Filter Mixer, Path Rendering Unit, and Allpass Filter Renderer are 487.34 kbytes, 281.25 kbytes, and 37.69 kbytes, respectively.

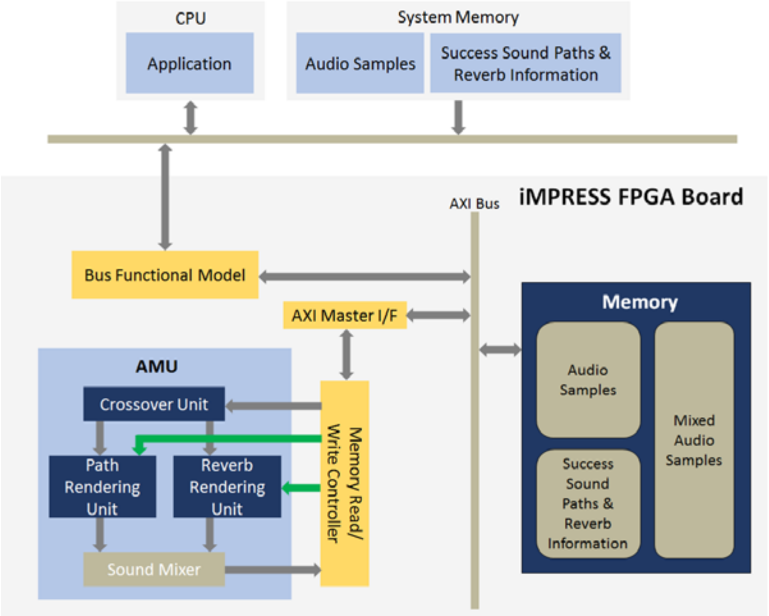


Fig. 13. FPGA prototype architecture implemented with an Artix-7 XC7A200T FPGA chip.

Table 2. Hardware compelxity: the number of floating-point arithmetic elemetns per FPGA chip

	MUL	2-input adder	3-input adder	SRAM size
Crossover Unit	60	-	24	-
Reverb Rendering Unit	168	110	28	525.03 kbytes
Path Rendering Unit	20	-	26	281.25 kbytes
Sound Mixer	-	-	1	-
Total	248	110	78	806.28 kbytes

5. Experimental Results and Analysis

In this section, the benchmark for evaluating the quality and the performance of sound is described. After that, we analyze and describe the results of the proposed AMU hardware architecture on an FPGA. Lastly, we present the user evaluation result of the downmixed audio to validate the quality of sound for our downmixed audio.

5.1 Benchmarks

We created 24 benchmarks by combining six 5-channel audio samples (left, right, center, left surround, and right surround), 2 scenes, and 2 materials (see **Table 3**) for the performance and the user evaluation. The audio samples were extracted from DVD movies. Sound propagation paths used for the experiment were generated by pre-processing the SPF for each scene model. The format of the sound source is signed PCM 16-bit and the sample rate of the audio sample is 48,000 Hz.

Fig. 14 shows the scene models and material used in the experiment. We set homogeneously the surface material of the scenes so that, depending on the material, we could experience different sound effects. The concert hall model (about 13,000 triangles) is modeled assuming the actual concert hall, that allows indirectly the real experience of our downmixed sound. Sibenik Cathedral (about 80,000 triangles) is a popular model for ray tracing. This can give an opportunity to user experience a relatively long late reverberation effect compared to other low-triangles models.

Table 3. The benchmark for performance evalution and user validation

Item	Description
Audio Sample	Gladiator, the Lost World: Jurassic Park, Chicken Run, the Lord of Rings: The Fellowship of the Ring, U-571, Steely Dan: Cousin Dupree
Scenes	Concert Hall, Sibenik Cathderal
Materials	Gypsum Board, Concrete

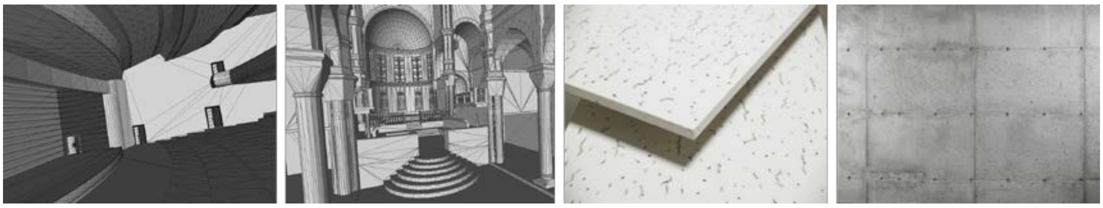


Fig. 14. The scene models and materials: Concert Hall, Sibenik, Gypsum Board, and Concrete

5.2 Audio Downmixing Performance of the FPGA Prototype

A quantitative performance analysis of the AMU is described in this section. The performance results measured using our FPGA prototype are shown in **Table 4**. This table includes the number of supportable sound sources/ paths, processing time, sample per second, frame per second (FPS), and FPGA utilization.

In terms of performance, our AMU can process 10 sound propagation paths and up to 5 sound sources at real-time rates. This result satisfies the constraint for a large number of sound sources proposed in [7]. In terms of performance scalability, if implemented in multicore AMU systems, the performance of our AMU can be linearly increased. The reason is that our AMU is designed with a fully fixed pipeline, which has simple architecture and high throughput capacity models.

In the view point of resource utilization, our AMU occupies 33%, 56%, 6%, and 17% of the Artix-7 resources LUT, Block RAM (BRAM), FF and DSP respectively. For considering the AXI interface and the bus system, it is possible to configure single AMU per Artix-7 XC7A200T FPGA chip. BRAM utilization is 56% because the PRU, Channel Allpass Filter Renderer, and Channel Comb Filter Renderer have a delay buffer independently for late reverb effect and delayed sound effect.

Table 4. Experimental result of the proposed architecture

Item	Description	
Number of supportable sound sources	Max. 5 sound sources per AMU	
Number of supportable sound propagation paths	Max. 10 paths per sound source	
Processing Time (# of sound sources: 5, # of length: 10 sec.)	192.01 ms (2.5M sample per sec, 52.08 FPS)	
FPGA Utilization (AMU only)	LUT Utilization	33 %
	BRAM Utilization	56 %
	FF Utilization	6 %
	DSP Utilization	17 %

The power consumption of the FPGA chip [30] was measured with the XPE. To measure power consumption, the variables should be input at least six: the number of LUTs/ BRAMs/ FFs/ DSPs, temperture condition that chip operates, environment option, and etc. We can also choose from six temperture conditions (typical: 25°C/ 50°C/ 75°C, maximum: 85°C/ 100°C/ 125°C) and three environments (250 LFM, 250 LFM w/ Heatsink, still air)

We measured separately the power consumption of the AMU into two conditions and environments: normal case and worst case. We conducted experiments with only the AMU except the AXI memory controller and bus interface. Since the maximum condition allowed in Artix-7 XC7A200T is 100°C, we chose the worst-case condition as 100°C. In the **Table 5**, the AMU consumes 1421 mW and 382 mW of power in the worst case and the normal case, respectively. Dynamic power of core appears to be the same in both cases, which seems to be due to the same LUT, BRAM, FF, and DSP utilization in both cases.

Table 5. The result of power estimation using XPE

Item		Normal Case	Worst Case
Conditions		Typical, Ambient=50°C	Maximum, Junction=100°C
Environment		Still Air	Still Air
Result	Core Dynamic Power	183 mW	183 mW
	Device Static Power	199 mW	1238 mW
	Total On-Chip Power	382 mW	1421 mW

5.3 User Evaluation

In this section, results from user evaluation studies which evaluate the sound quality of downmixed audio generated by the AMU are presented. We describe the study design and procedure, and then analyze the sound quality of the AMU based on results of the user evaluation.

A. Study Design

To measure the sound quality of the AMU-mixed audio, we conducted the user evaluation to compare our results with the downmixed audio by the commercial tool. The commercial tool used for comparison was PowerDVD (CyberLink). PowerDVD offers 3 downmix options. Among them, TrueTheater Surround is a virtual surround technology developed by CyberLink that offers various surround options such as stadium, theater, living room, etc. [31]. We call the AMU-mixed audio as "AMU_MIX" and the audio generated by PowerDVD as "PowerDVD_MIX".

The questionnaires used in the user evaluation is shown in **Table 6**. The questionnaire consisted of 8 items in total, with five VR related items denoted as VR No. 1-5 and three 3D sound effect related items. The evaluation criterion for all items is 5 examples, that is from "strongly disagree" as 1 to "strongly agree" as 5.

Table 6. The contents of our survey

No.	Contents
VR No. 1	I have experienced VR
VR No. 2	I often enjoy VR.
VR No. 3	If the market of VR is activated, I will play the contents of VR actively.
VR No. 4	It satisfies the sound effect level of the VR contents which is currently released.
VR No. 5	In order to enjoy VR contents, it is necessary to have excellent sound effect.
3D Audio No. 1	The difference between when applying the method proposed by the demo and when not applying it is remarkable.
3D Audio No. 2	The proposed audio downmixing sound effect is better than the Power DVD sound effect.
3D Audio No. 3	I think that the proposed audio downmixing sound effect improves the quality of the contents.

VR No. 1-3 are prepared to assess the degree of interest of the subjects in VR, and VR No. 4-5 are to assess how important the sound effect is in the VR content of subject. No. 1 in the 3D sound effects item is a question to assess whether the subject felt the sound effect of AMU_MIX. No. 2 is a direct comparison of PowerDVD_MIX and the sound effects of AMU_MIX. No. 3 is a question about whether the AMU_MIX sound effect enhances the overall content quality when the subject feels it.

B. Study Procedure

Ten subjects consisting of one female and nine males participated in the user evaluation and they consists of the following three groups: three ordinary users, two persons with high interest in the sound system, and five audio engineers. All subjects have normal hearing and they were between 27 and 47 years old, the average age was about 31 years old. Before performing the listening assessment, we first conducted a questionnaire survey on VR related questions to figure out the degree of interest in the subject's VR and the importance of the sound effect in the VR environment.

We have prepared a benchmark demo that allows the subjects to change the movie clip, scene, material, and rendering mode in directly. Before playing to the demo, they were given a full explanation of the sequence of the experiments and how to operate the demo. They watched the demo using the prepared stereo speakers and headphones. When asked about the demo manipulation during appreciation, we paused the demo and explained it again.

Table 7. The results from the our survey of 10 subjects

No.	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Average	Standard deviation
VR No. 1	-	-	-	7	3	4.3	0.48
VR No. 2	1	2	5	1	1	2.9	1.10
VR No. 3	-	-	3	3	4	4.1	2.62
VR No. 4	-	6	4	-	-	2.4	0.52
VR No. 5	-	-	1	3	6	4.5	0.71
3D Audio No. 1	-	-	-	-	10	5.0	0.00
3D Audio No. 2	-	-	2	4	4	4.2	0.79
3D Audio No. 3	-	-	-	6	4	4.4	0.52

C. User Evaluation Results

Table 7 and **Fig. 15** summarized the result of our survey. For VR No. 1-3 items, the answers to the questions were "agree" and "strongly agree" with results of 36% and 27%, respectively. As a majority of the subjects has interests in VR, the meaning of the result can be deduced. The results for VR No 4-5 shown in **Fig. 15 (b)** show that the subjects have made a negative response of 30% and a positive response of 45% for the importance of sound effects.

Fig. 15 (c) shows the results for the 3D sound effects questions. After experiencing the demos which we prepared, all subjects replied that the difference in AMU_MIX application was noticeable (see the first graph in **Fig. 15 (c)**). The comparison of quality of audio downmix between PowerDVD_MIX and AMU_MIX is shown in the second graph of **Fig. 15 (c)**. Eight of the subjects responded “strongly agree” or “agree” with better quality of AMU_MIX than PowerDVD_MIX. The other two respondents replied that the effect of PowerDVD_MIX’s 3D audio is better than AMU_MIX, but their preference is closer to PowerDVD_MIX.

Finally, as for the sound effect of AMU_MIX, the results of the user evaluation showed that the subjects were positive about applying the proposed system to the contents (see the third graph of **Fig. 15 (c)**). Subjects who negatively thought about VR No. 4-5 were found to be aware of the importance of sound effects in VR contents through demo experience. Through these results, we found that the proposed system improves the sound effect of VR contents.

6. Conclusion

For immersive realistic virtual reality, virtual space should reproduce acoustic expression as well as visual one. The importance of reproducing the acoustic spaciousness for users to experience more realistic VR is increasing because auditory cues can enhance the perception of the complex surrounding environment without the visual system. Based on sound rendering for mobile platforms, this paper proposed a hardware-based real-time audio downmixing system.

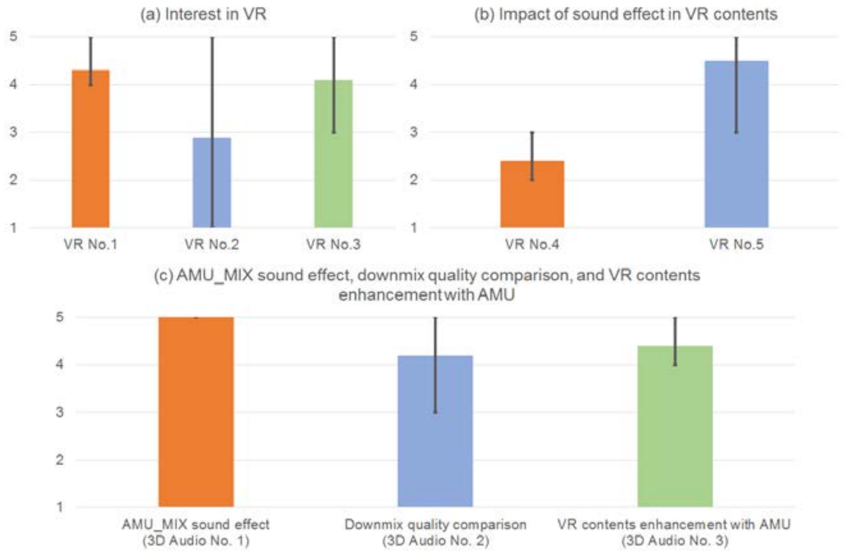


Fig. 15. The user evaluation results of our survey

On an FPGA chip, we implemented the proposed system. Through the performance evaluation on the FPGA prototype, we verified that the proposed system achieved low power consumptions and a real-time rate. In order to examine the quality of the proposed method, we performed the user evolution and found that it showed good quality by comparison with the commercial tool.

The proposed system has several limitations. Our AMU uses a SRAM with a large delay buffer, relatively. For this reason, our AMU limits the delay time of the PRU up to 25ms. Our AMU is designed in a fully fixed pipeline manner. If the contents developer wishes to separate the frequency band range used by the AMU, there is an issue that needs to be modified for the entire hardware architecture.

The future work will include a delay buffer size reducing algorithm would be devised to reduce chip area and power consumption. And it will include standalone sound rendering hardware solutions by integrating the proposed hardware with dedicated hardware [32] that accelerates the sound propagation stage. Finally, we will conduct ASIC evaluation on the standalone solution to measure performance, chip area and power, and then study on the possibility of mobile platform application.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. 2016-0-00204, Development of mobile GPU hardware for photo-realistic realtime virtual reality). The CAD tools were supported by IDEC.

References

[1] K. S. Hale and K. M. Stanney (ed.), “Handbook of virtual environments: Design, implementation, and applications,” *CRC Press: Boca Raton*, US, 2014. Article (CrossRef Link)

[2] M. Vorlander, “Auralization: Fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality,” *Springer: Berlin*, Germany, 2008.

[3] A. Febrettia, A. Nishimotoa, T. Thigpena, J. Talandisa, L. Longa, J. Pirtlea, T. Peterkaa, A. Verloa, M. Browna, D. Plepysa, D. Sandina, L. Renambota, A. Johnsona, and J. Leigha, “Cave2: a hybrid reality environment for immersive simulation and information analysis,” in *Proc. of Proceedings of SPIE Electronic Imaging 2013,* Burlingame, CA, USA, 3-7 Feb. 2013. Article (CrossRef Link)

[4] L. Antani, N. Galoppo, A. Lake, and A. Peleg, “Next-gen sound rendering with OpenCL,” in *Proc. of Proceedings of ACM SIGGRAPH 2011 BOF OpenCL*, Vancouver, BC, Canada, 7-11 Aug. 2011.

[5] B. Hook and T. Smurdon, “An intro to virtual reality audio,” in *Proc. of Game Developers Conference (GDC) 2015,* San Francisco, CA, USA, 2-6 Mar. 2015.

[6] N. Ward-Foxton, “Environmental audio and processing for VR,” in *Proc. of Game Developers Conference (GDC) 2015*, San Francisco, CA, USA, 2-6 Mar. 2015.

[7] A. Chandak, “Effcient geometric sound propagation using visibility culling,” *Ph.D. dissertation*, Department of Computer Science, University of North Carolina at Chapel Hill, NC, USA, 2011.

[8] S. Borkar and A. A. Chien, “The future of microprocessors,” *Communications of the ACM*, vol. 54, no. 5, pp. 67-77, May 2011. Article (CrossRef Link)

[9] Dolby, “Dolby ATMOS,” Article (CrossRef Link)

[10] B. V. Daele and W. V. Baelen, “Professional workflow white paper,” Auro Technologies NV, 2011.

[11] AM3D, “ZIRENE 3D: Positional audio,” Article (CrossRef Link)

[12] Khronos, “OpenSL ES 1.1 specification,” Article (CrossRef Link)

[13] Oculus, “Oculus audio SDK guide,” Article (CrossRef Link)

[14] M. T. Taylor, A. Chandak, L. Antani, and D. Manocha, “RESound: Interactive sound rendering for dynamic virtual environments,” in *Proc. of Proceedings of the 17th ACM international conference on Multimedia (MM 2009)*, Beijing, China, 19-24 Oct. 2009. Article (CrossRef Link)

[15] N. Raghuvanshi, J. Snyder, R. Mehra, M. Lin, and N. Govindaraju, “Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes,” *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 68:1-11, Jul. 2010. Article (CrossRef Link)

[16] R. Mehra, N. Raghuvanshi, L. Antani, A. Chandak, S. Curtis, and D. Manocha, “Wave-based sound propagation in large open scenes using an equivalent source formulation,” *ACM Transactions on Graphics*, vol. 32, no. 2, pp. 19:1-13, Apr. 2013. Article (CrossRef Link)

[17] R. Mehra, A. Rungta, A. Golas, M. Lin, and D. Manocha, “WAVE: Interactive wave-based sound propagation for virtual environments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 434-442, Jan. 2015. Article (CrossRef Link)

[18] M. Vorlander, “Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm,” *The Journal of the Acoustical Society of America*, vol. 86, no. pp. 172-178, 1989. Article (CrossRef Link)

[19] C. Schissler and D. Manocha, “Interactive sound propagation and rendering for large multi-source scenes,” *ACM Transactions on Graphics*, vol. 36, no. 1, pp. 2:1-12, Feb. 2017. Article (CrossRef Link)

[20] C. Cao, Z. Ren, C. Schissler, D. Manocha, and K. Zhou, “Interactive sound propagation with bidirectional path tracing,” *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 180:1-11, Nov. 2016. Article (CrossRef Link)

[21] Nvidia, “NVIDIA VRWorks audio,” Article (CrossRef Link)

[22] AMD, “AMD TrueAudio Next (TAN),” Article (CrossRef Link)

[23] Valve, “Steam Audio,” Article (CrossRef Link)

[24] C. Schissler and D. Manocha, “GSound: Interactive sound propagation for games,” in *Proc. of Proceedings of Audio Engineering Society 41st International Conference: Audio for Games*, London, UK, 2-4 Feb. 2011.

[25] S. Butterworth, “On the theory of filter amplifiers,” *Experimental Wireless and the Wireless Engineer*, vol. 7, no. 6, pp.536-541, Oct. 1930.

[26] B. Luo, W. Shichuang, and Z. Nanrun, "Flexible design method for multi-repeater wireless power transfer system based on coupled resonator bandpass filter model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 11, pp. 3288-3297, Nov. 2014. Article (CrossRef Link)

[27] G. Garcia, “Optimal filter partition for efficient convolution with short input/output delay,” in *Proc. of 113th Convention of the Audio Engineering Society*, Scotts Valley, CA, 5-8 Oct. 2002.

[28] M. R. Schroeder, “Natural sounding artificial reverberation,” *Journal of the Audio Engineering Society*, vol. 10, no. 3, pp.219-223, Jul. 1962.

[29] Xilinx, “Xilinx 7 Series FPGAs Data Sheet: Overview,” Article (CrossRef Link)

[30] Xilinx, “Xilinx Power Estimator: 7 Series and Zynq-7000 Ver. 2017.02,” Article (CrossRef Link)

[31] CyberLink, “CyberLink PowerDVD17 User Guide,” Article (CrossRef Link)

[32] D. Hong, T.-H. Lee, Y. Joo, and W.-C. Park, “Real-time sound propagation hardware accelerator for immersive virtual reality 3D audio,” in *Proc. of Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D 2017)*, San Francisco, CA, USA, 25-27 Feb. 2017. Article (CrossRef Link)

[33] The Wall Street Journal, “Augmented and Virtual Reality: A New Vision,”Article (CrossRef Link)

Dukki Hong received the B.S. and Ph.D. degrees in Computer Engineering from Sejong University, Korea, in 2011 and 2018, respectively. His current research interests include graphics hardware architecture, high performance computer architecture, real-time sound rendering, real-time ray tracing, embedded systems, system-on-chip design, lossless image compression, parallel processing algorithms.

Hyuck-Joo Kwon received the B.S., M.S., PhD. degrees in Computer Engineering from Sejong University, Korea in 2009, 2011, and 2016, respectively. Currently, he is a postdoctoral researcher in Electrical and Electronic engineering, Yonsei University, Korea. His research areas include hardware acceleration, 3-D rendering processor architecture, mobile GPU and real-time ray tracing.

Cheong Ghil Kim received the B.S. in Computer Science from University of Redlands, California in 1987, and the M.S. and Ph.D. degree in Computer Science from Yonsei University, Korea, in 2003 and 2006, respectively. He is a professor of Computer Science, Namseoul University, Seoul. His current research interests include Mobile AR, Multimedia Embedded Systems, and 3D Contents.

Woo-Chan Park received the B.S., M.S., and Ph.D. degrees in Computer Science from Yonsei University, Korea, in 1993, 1995, and 2000, respectively. From 2001 to 2003, he was a Research Professor with Yonsei University. He is a professor of Computer engineering, Sejong University, Seoul. His current research interests include real-time ray-tracing rendering, computer arithmetic, advanced computer architecture, and lossless image compression architecture.