Sounds Real: Using Hardware Accelerated Real-time Ray-Tracing for Augmenting Location Dependent Audio Samples

Alexander Madzar
TU Kaiserslautern
Kaiserslautern, Germany
madzar@rhrk.uni-kl.de

Linfeng Li
TU Kaiserslautern
Kaiserslautern, Germany
lli@rhrk.uni-kl.de

Hymalai Bello
DFKI
Kaiserslautern, Germany
hymalai.bello@dfki.de

Bo Zhou
TU Kaiserslautern and DFKI
Kaiserslautern, Germany
bo.zhou@dfki.de

Paul Lukowicz
TU Kaiserslautern and DFKI
Kaiserslautern, Germany
paul.lukowicz@dfki.de

***Abstract*—We present a data augmentation technique for generating location variant audio samples using ray-traced audio in virtual recreations of the real world. Hardware Audio-Based Location-Aware Systems are capable of locating audio sources in relation to mobile devices. This is a relevant technique in the context of location-based and person tracking in ubiquitous environments. However, this solution is limited in collecting vast data to train the machine learning model reliably. To overcome this problem, we constructed a virtual environment using the audio ray-tracing solution, NVidia VRWorks Audio in Unreal Engine 4, to simulate a real-world setting. The environmental sounds in the real-world scenario were imported into the virtual environment. This strategy could augment data for training Hardware Audio-Based Location-Aware Systems machine learning models with the necessary calibration of the unreal and real data sets. Our results show the audio ray-tracing framework could simulate real-world sound in the virtual environment to a certain extent.**

***Index Terms*—audio synthesis; ray-tracing; Unreal Engine; VRWorks audio.**

I. Introduction

Virtual Environments (VE) have steadily increased in popularity among researchers over the last decade [1][2]. VE can give an immersive simulation experience that is suitable to a variety of use cases, including gaming, automobile, construction, and education [3][4][5]. Apart from graphics, audio processing is a vital component of the virtual environment's immersive experience. Numerous ray-tracing audio frameworks, such as VRWorks Audio (NVIDIA) [6], or Steam Audio [7], can enhance and increase the realism and immersion impact in virtual reality technologies.

Sound is an interesting tool to localize users and mobile devices. For example, the location of a user walking within a building can be determined based on ambient sounds [8]. In this scope, we define such systems as Hardware Audio-Based Location-Aware Systems (HABLAS). In ubiquitous computing, sound can be a helpful sensing modality to classify location [9][10].

However, hardware-based location solutions have limitations in terms of data collection. For example, in a particular scene and combination of environmental sounds, the data can only be collected where the devices are located. If the opportunity has passed, one cannot try another location and recollect the data with that exact scene and environment sounds. This is on top of the already well-known general difficulty in gathering annotated data.

In this work, we proposed to use virtual environments created by game design engines via audio path tracing frameworks to generate location-dependent sound data for training HABLAS machine learning models. A critical remark of our proposal is that the quality of the augmentation of virtually generated sound will have a strong dependency in the calibration with the real-sound and the data augmentation algorithm selected for the task.

Our paper structure is as follows; Section II presents related work in the areas of Virtual Environment (VE) reconstruction and Real-Time ray-tracing solutions. Next, Section III provides a detailed description of the proposed method, including details of the employed framework, and the interface between the real-world sound recording with the VE. Then, in Section IV a spectral comparison between the recorded real-world sounds with the virtually generated sounds is discussed. Finally, in Section V, we conclude our work and discuss further ideas.

II. Related Work

With the growing popularity of Virtual Reality (VR), NVidia has released a software suite VRWorks for VR and game developers to utilize graphics processing units (Audacity Team) acceleration with existing 3D design environments [11]. Such tools have also received attention from researchers, especially in the visual scene reconstruction discipline, in works, such as [12][13][14].

However, traditional graphical rendering struggles to achieve the real-world effect, mainly since, in reality, what we perceive results from light and sound waves bouncing off different surfaces and reaching our eyes and ears. Ray-tracing solves such a problem as the graphics or audio can be rendered not only by geometry, but also by considering the reflection and refraction of simulated rays closer to real life. Real-time

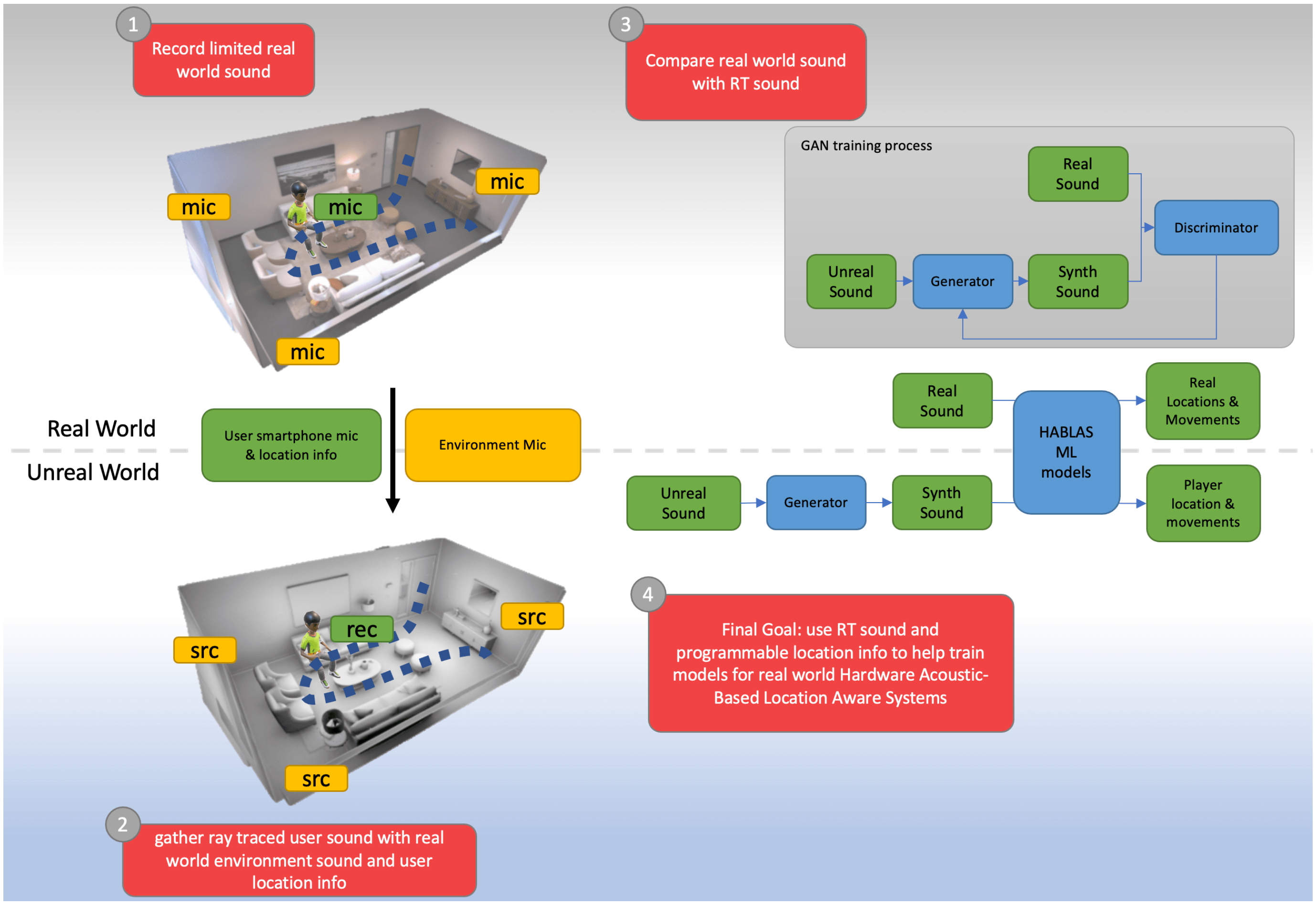
1
Record limited real world sound
3
Compare real world sound with RT sound
GAN training process
Real Sound
Discriminator
Unreal Sound
Generator
Synth Sound
mic
mic
mic
mic
Real Sound
HABLAS ML models
Real Locations & Movements
Real World
Unreal World
User smartphone mic & location info
Environment Mic
Unreal Sound
Generator
Synth Sound
Player location & movements
src
rec
src
src
4
Final Goal: use RT sound and programmable location info to help train models for real world Hardware Acoustic-Based Location Aware Systems
2
gather ray traced user sound with real world environment sound and user location info

Figure 1. Illustration of our approach to generate location dependent sound samples from virtual recreations of real-world scenes.

ray-tracing has been difficult due to the extreme computation demand [15]. Most recently, real-time ray-tracing has also become possible thanks to the latest GPUs with dedicated accelerators. VRWorks has also been updated with accelerated ray-tracing. GPU acceleration has been investigated for ray-traced sound propagation in underwater environments [16].

Additionally, the idea of using virtual scenes to generate synthetic visual training data as input to machine learning methods has been explored in [17], where Unreal Engine 4 [18] was used to generate and automatically annotate ground truth data of robot agents interacting with objects and between each other. Moreover, a commercial solution for simulation and training of artificial intelligence (AI) robotics was introduced by NVIDIA®Isaac™[19], in which synthetic and virtual data generation techniques can improve the behavior of the robot.

Therefore, we could argue that VE are becoming more similar to the real-world, at least to some degree. The above led us to benefit from virtual scenes recreation for the generation of synthetic data based on ray-traced sound for data augmentation in machine learning solutions, which to the best of our knowledge, has not been done before.

III. Method Proposal

Ray-tracing audio solutions, such as VRWorks Audio (NVidia) and Steam Audio can merge path tracing effects, such as sound propagation, reflection, and constructive/destructive collisions between different sound sources. As a result, and in conjunction with the influence of the various materials' sound properties (e.g., absorption, reflection, transmission, etc.), those solutions augment the immersion experience to the user. For example, a hallway with carpets and wooden walls would sound differently from the same hallway with surrounding marble materials. These frameworks are compatible with game development engines like Unreal Engine 4. This research aims to prove the feasibility of audio data augmentation using the VRWorks Audio virtual reality ray-tracing audio framework. To test our approach, we have created a virtual environment in Unreal Engine 4 modeled after a physical, real-world location (meeting room), as shown in Figure 2, to generate a baseline design with the potential to mimic several users (avatars) in different social circumstances. Our particular focus was to record sound data in the real scene and to compare it with captured sound in the virtual scene designed in the Unreal Engine. Therefore, after calibrating the unreal and real data

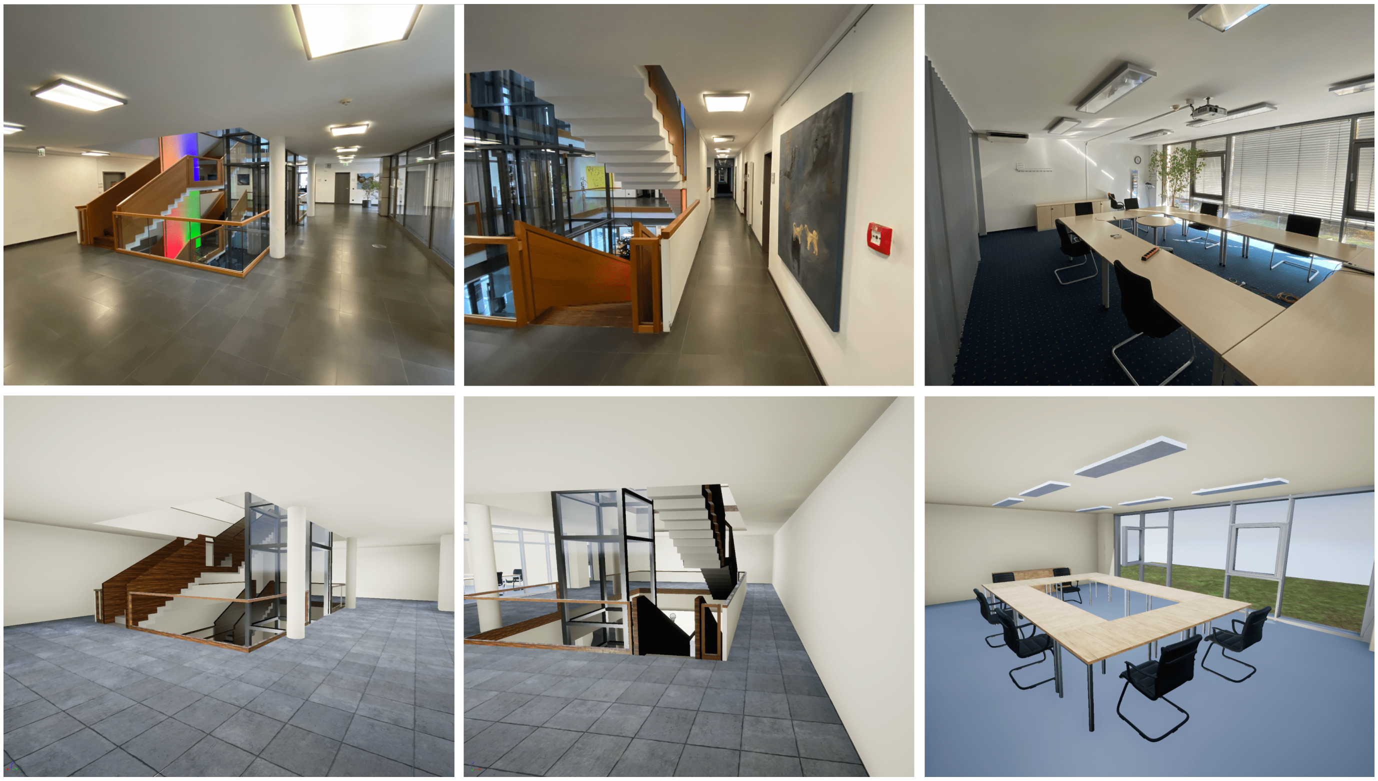

Figure 2. The real scene in the top row and the reconstructed (Unreal Engine) scene in the bottom row.

sets, this strategy could be employed to generate synthetic data for machine learning models to be used in HABLAS solutions, as depicted in the research project's process in Figure 1.

Figure 1 shows a block diagram of the main idea; first, audio recordings from a real scenario will be collected in defined positions. Secondly, the real sound sources are played inside the VE, and the listener is set in the predefined positions in the first step. Next, a comparison between the real sound and the virtually generated sound is made, and a Generative Adversarial Network (GAN) specifically designed for audio [20][21][22] could be used as the data augmentation technique. Finally, these augmented location-aware sounds are used to tackle the lack of training data in models for HABLAS.

A. Frameworks and Methodology

We used Unreal Engine 4 (UE4) to model the scene setup in combination with Nvidia VRWorks Audio. More precisely, Unreal Engine 4.15 is the version for which VRWorks Audio is available as a plugin [23]. The plugin is a private repository; therefore, an Unreal developer with a personal Github account is needed to enable the link. UE4 is originally a game engine for game development but has now been adopted by various other industries. With UE4, a desired scene can be created relatively quickly in a modular fashion. Furthermore, VRWorks Audio enables immersive audio through ray-traced sound in 3D space in real-time. The key features of VRWorks Audio [24] include effects, such as sound propagation, occlusion for direct and indirect paths, attenuation, material reflection, absorption, and transmission, which are needed for ray-traced sound. In the following, we will present how we created the scene setup in Unreal and VRWorks. The software suite was tested on a Dell XPS15 laptop with an NVidia GTX 1050 GPU, as well as two workstations with an NVidia GTX 1080 and RTX A6000 GPUS. All systems can render the scene with real-time performance.

B. Real World Recordings

A real-world experiment was designed to record ambient sound at predetermined places and simulate the static positions of the avatar inside Unreal Engine. We recorded the meeting room's ambient sound using eleven iOS devices (iPhone and iPad). As shown in Figure 3, devices 1, 2, 3, 4, 6, 7, and 11 were arranged on the tables. The microphone symbols were used for gathering environment sound assets to play during real-time rendering, and the numerical symbols are used to distinguish sound differences at different locations. Device 5 was located on the cabinet on the left side of the meeting room. To create some distinctions for sound recording, we left the left door closed and placed devices 8 and 10 in front of the left and right doors, respectively. Device 9 was placed in front of the elevator in the hallway. The entire recording took almost two hours. During the time, various activities were induced at random places in the scene, such as moving trolleys in the corridor, multiple people walking, powered drills, hammering metal, conversations, music, etc. The windows face a busy road, and there is always traffic

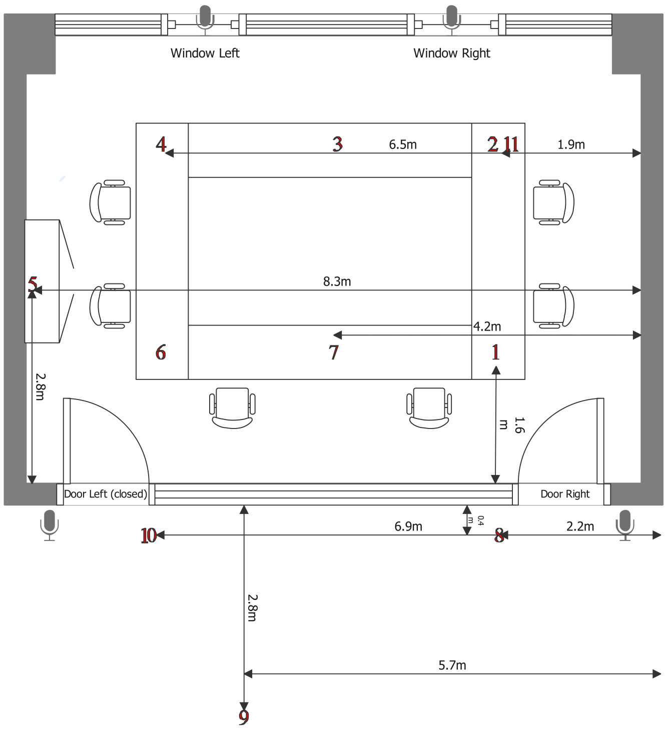

Figure 3. Floor plan: placement of mobile recording devices inside the real/virtual meeting room.

sound captured at the two microphones at the windows. The four microphones depicted in Figure 3 were used as sound sources inside the real-world. They were placed on the doors and windows, and all of them possess unidirectional cardioid polar patterns, which means that most audio with an incident angle outside 90-270 degrees will be attenuated [25]. Android devices recorded such microphones at 48Khz with 256kbps. To reduce the effect of the wind, they were covered with a fur windshield.

C. Sound Recordings of Unreal World

In the Unreal World, the static locations described in Section III-B were used to perform the augmentation of the sound recordings coming from the iOS devices. First, the recordings of the four ambient sound microphones at the windows and doors were imported into Unreal Engine. They were then used as audio sources at the exact corresponding locations. Next, the sound sources needed to be calibrated according to the following considerations:

- **Attenuate:** Enables the sound attenuation. If false is specified, the sound will play at maximum volume regardless of the distance between the sound source and the listener. The setting must be set to true.
- **Spatialize:** This property enables spatialization, which denotes the projection and localization of sound sources inside the virtual environment. If set to false, the sound will be non-spatialized, and no panning will be applied when the listener moves around.
- **Distance Algorithm:** Five different distance algorithms are available, which determine the attenuation rate over distance. We experimented with all the functions and, by ad-hoc method, concluded that the *Inverse*, *Natural Sound* and *Logarithmic* distance algorithms realistically simulate our experimental sound.
- **Attenuation Shape:** This property specifies the shape used to establish the sound's minimum and maximum attenuation points. There are four alternative shapes available, with *Sphere* being the default setting and producing a spherical attenuation shape. The spherical form is the most accurate representation of how sound propagates in the real-world [26].
- **Radius:** The radius describes the distance from the location of the sound at which the falloff begins. Our radius was set to 50 cm.
- **Falloff Distance:** This describes the distance over which the falloff ends. We experimented with several values and got the most realistic results with a falloff distance of 2500 cm.
- **Occlusion:** Occlusion is disabled as ray-tracing is the more realistic alternative.
- **Direct Path Gain:** We use the default setting from the VRWorks Audio Tutorial [6]. Set the value to 5.0.
- **Indirect Path Gain:** Here, we also take the default setting, set the value to 5.0.
- **Effect Strength:** We also use the same preset as in the tutorial: *High*.

To conduct the recordings in the Unreal World, the avatar was placed at the eleven static locations that are presented in Figure 3, so that we can compare them with the real-world recordings from iOS devices. This means that we placed the game character (the listener) at locations one to eleven. The complete recorded sounds from the sound sources (4 microphones, doors, and windows) were reproduced every time the avatar changed location. The sound from the audio sources inside the virtual environment was recorded for about two hours. The recordings were made using the audio software *Audacity*® [27]. Audacity® software is copyright © 1999-2021 Audacity Team. The name Audacity® is a registered trademark. The operating system's digital output was looped back to the recording software.

IV. Comparison Real and Unreal Recordings

After recording in Unreal Engine 4 to determine the performance of Nvidia's ray-tracing technology, a user perception assessment was performed. Our initial test was to have four persons as the audience to only listen to the audio, while one developer navigates the avatar in the real-time rendered scene in various locations. The audience would then describe the change of location. Two audiences tested with the laptop's speaker, and the other two audiences tested via Zoom meetings. The initial test result has revealed that it is evident whether the avatar is in the meeting room, close to the window, or the hallway outside the meeting room. They describe their major clues were the traffic sound outside the window and

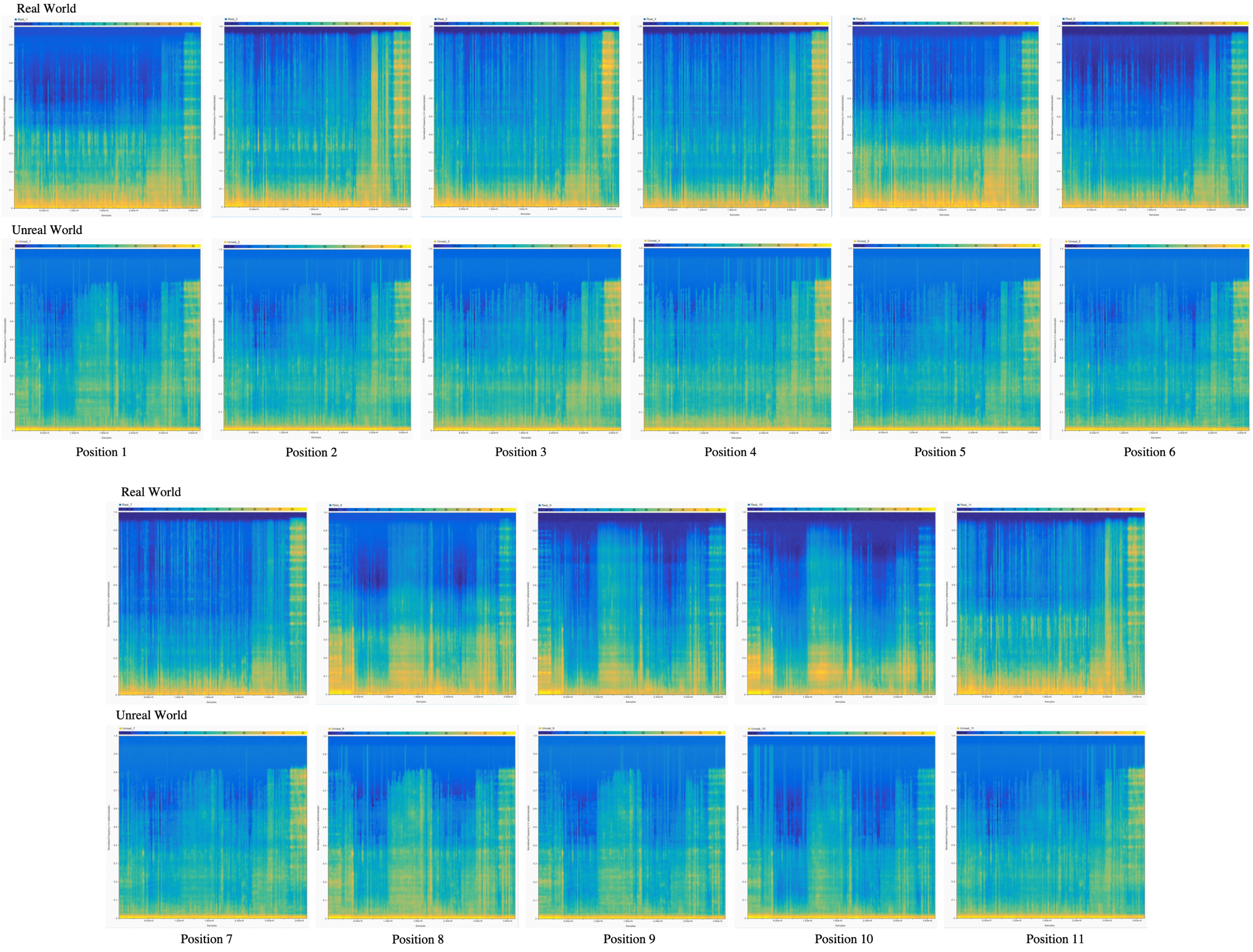

Figure 4. Spectrogram comparison between real scene and the reconstructed (Unreal) scene setup at various locations, during exactly the same time.

the ambient echo characteristic. The hallway with ceramic tile floors and emptier space sounds significantly different than the smaller furnished meeting room with carpet floors.

Next, we present the differences between the real and unreal worlds at different predefined locations with the same time moments using the spectrograms in Figure 4. First, we cropped identical 90-second sections from the 2-hour recordings for each position. We then converted these sections into spectrograms using the Matlab ®Signal Analyzer App [28] to evaluate the differences or similarities between the real and unreal recordings.

When examining the spectrograms of the different positions, it is noticeable that each position has a distinct appearance, although the sound played was the same in all 11 positions, as expected. It is indicating that the localization by the ray-tracing is valid. Depending on the position in the room, the sound is influenced by the arrangement of the sound sources, surrounding objects like chairs and tables, and the materials used. By comparing the spectrograms of the real and virtual environment worlds, it is clear that the virtual environment world's spectrograms are less detailed in the high-frequency range compared to the real-world. This cut-off of high frequencies is most likely due to the simulation environment, i.e., software restrictions, resolution of the digital sound card on the computer.

Additionally, certain events in the real-world (sections with intense yellow colors) can be noticed on the spectrograms of the virtual environment world in Figure 4. For instance, three events stand out at positions 8-10. These are construction activities that occurred in the hallway adjacent to these positions. However, these events are not discernible from the remaining locations, as construction work was barely audible within the meeting room.

Although in our work the virtual scene is manually reconstructed based on real-world measurements, there are already trends towards automatic scene reconstruction, such as [29].

V. Conclusion and Future Work

Overall, we have investigated sound sample augmentation using existing software tools meant for 3D game and VR experience developers for data augmentation, specifically in location-dependent sound samples. This can be used further for training Hardware Audio-Based Location-Aware Systems, bypassing difficulties in data collection. Modern tools utilize hardware-accelerated ray-tracing for the audio rendering, thus making it more realistic than traditional propagation-only synthesis.

A significant advantage of our method is that the scene can be replayed, and the listener can be placed at different locations for the same period of surrounding activities, which is not possible in real-world data collection for this purpose. In this work, although we only presented the locations that were also recorded in the real-world in Figure 4, our approach can generate a new soundtrack from any location other than those inside the scene. It is impossible for actual data collection procedures to gather data at every centimeter inside the scene.

In our future work, we would continue to implement our method with SteamVR, which also has ray-traced sound. Furthermore, we will also improve the generated sound samples using generative adversarial networks as a post-processing step.

Acknowledgment

This work has been supported by BMBF (German Federal Ministry of Education and Research) in the project SocialWear.

References

[1] Y. Lang, L. Wei, F. Xu, Y. Zhao, and L.-F. Yu, “Synthesizing personalized training programs for improving driving habits via virtual reality,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 297–304.

[2] H. Zhang, “Head-mounted display-based intuitive virtual reality training system for the mining industry,” *International Journal of Mining Science and Technology*, vol. 27, no. 4, pp. 717–722, 2017.

[3] M. Thees, S. Kapp, M. P. Strzys, F. Beil, P. Lukowicz, and J. Kuhn, “Effects of augmented reality on learning and cognitive load in university physics laboratory courses,” *Computers in Human Behavior*, vol. 108, p. 106316, 2020.

[4] S. Kapp *et al.*, “Augmenting kirchhoff’s laws: Using augmented reality and smartglasses to enhance conceptual electrical experiments for high school students,” *The Physics Teacher*, vol. 57, no. 1, pp. 52–53, 2019.

[5] M. P. Strzys *et al.*, “Physics holo. lab learning experience: using smartglasses for augmented reality labwork to foster the concepts of heat conduction,” *European Journal of Physics*, vol. 39, no. 3, p. 035703, 2018.

[6] A. Dantrey and T. Scudiero. (2018) Creating immersive audio effects in games and applications using vrworks audio. NVIDIA. [Online]. Available: https://on-demand.gputechconf.com/gtc/2018/presentation/s8680-creating-immersive-audio-effects-in-games-and-applications-using-vrworks-audio.pdf

[7] Steam Audio. (2019) Steam audio unreal engine 4 plugin. Steam Audio. [Online]. Available: https://valvesoftware.github.io/steam-audio

[8] Y. Teshima, J.-y. Takayama, S. Ohyama, and K. Oshima, “Person localization using tdoa of non-speech sound signal based on multiplexed csp analysis,” in *Proceedings of SICE Annual Conference 2010*. IEEE, 2010, pp. 1207–1213.

[9] C. E. Galván-Tejada *et al.*, “A generalized model for indoor location estimation using environmental sound from human activity recognition,” *ISPRS International Journal of Geo-Information*, vol. 7, no. 3, p. 81, 2018.

[10] S. Park, S. Mun, Y. Lee, and H. Ko, “Acoustic scene classification based on convolutional neural network using double image features,” in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 2017, pp. 98–102.

[11] M. Kraemer. (2018) Accelerating your vr games with vrworks. [Online]. Available: https://www.gdcvault.com/play/1024356/Accelerating-Your-VR-Games-with

[12] O. Zia, J.-H. Kim, K. Han, and J. W. Lee, “360 panorama generation using drone mounted fisheye cameras,” in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–3.

[13] I. Baek, A. Kanda, T. C. Tai, A. Saxena, and R. Rajkumar, “Thin-plate spline-based adaptive 3d surround view,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 586–593.

[14] D. Pohl, N. Choudhury, and M. Achtelik, “Concept for rendering optimizations for full human field of view hmds,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 663–664.

[15] J. Gunther, S. Popov, H.-P. Seidel, and P. Slusallek, “Realtime ray tracing on gpu with bvh-based packet traversal,” in *2007 IEEE Symposium on Interactive Ray Tracing*. IEEE, 2007, pp. 113–118.

[16] M. Ulmstedt and J. Stålberg. (2019) Gpu accelerated ray-tracing for simulating sound propagation in water. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1352170/FULLTEXT01.pdf

[17] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano, and J. Garcia-Rodriguez, “Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation,” *Virtual Reality*, pp. 1–18, 2019.

[18] Epic Games. (2019) Unreal engine. Epic Games. [Online]. Available: https://www.unrealengine.com

[19] NVIDIA. (2019) Nvidia isaac platform for robotics. NVIDIA. [Online]. Available: https://www.nvidia.com/en-us/deep-learning-ai/industries/robotics/

[20] I. Goodfellow *et al.*, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[21] K. Kumar *et al.*, “Melgan: Generative adversarial networks for conditional waveform synthesis,” *arXiv preprint arXiv:1910.06711*, 2019.

[22] C. Donahue, J. McAuley, and M. Puckette, “Adversarial audio synthesis,” *arXiv preprint arXiv:1802.04208*, 2018.

[23] NvPhysX. (2017) Vrworks audio sample project. NVIDIA. [Online]. Available: https://github.com/NvPhysX/UnrealEngine/tree/VRWorks-Audio-4.15.1

[24] NVIDIA DEVELOPER. (2017) Vrworks - audio. NVIDIA DEVELOPER. [Online]. Available: https://developer.nvidia.com/vrworks/vrworks-audio

[25] A. Kuntz and R. Rabenstein, “Cardioid pattern optimization for a virtual circular microphone array,” in *EAA Symposium on Auralization*. Citeseer, 2009, pp. 1–4.

[26] Epic Games. (2019) Sound attenuation. Epic Games. [Online]. Available: https://docs.unrealengine.com/4.26/en-US/WorkingWithMedia/Audio/DistanceModelAttenuation/

[27] Audacity Team . (1999-2021) Audacity(r): Free audio editor and recorder. Audacity Team. [Online]. Available: https://audacityteam.org/

[28] MathWorks, *Matlab Signal Analyzer*, Natick, Massachusetts, United State, 2021. [Online]. Available: https://de.mathworks.com/help/signal/signal-analyzer.html

[29] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, “Atlas: End-to-end 3d scene reconstruction from posed images,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 2020, pp. 414–431.