Audio Engineering Society

Conference Paper

Presented at the AES International Conference on
Audio for Virtual and Augmented Reality
2020 August 17 – 19, Online

Cloud-Enabled Interactive Sound Propagation for Untethered Mixed Reality

Michael Chemistruck[1], Kyle Storck[1], and Nikunj Raghuvanshi[2]

[1]*Microsoft Mixed Reality*
[2]*Microsoft Research*

Correspondence should be addressed to Michael Chemistruck (michem@microsoft.com)

ABSTRACT

We describe the first system for physically-based wave acoustics including diffraction effects within a holographic experience shared by multiple untethered devices. Our system scales across standalone mobile-class devices, from a HoloLens to a modern smart phone. Audio propagation in real-world scenes exhibits perceptually salient effects that complement visuals. These include diffraction losses from obstruction, re-direction ("portaling") of sounds around physical doorways and corners, and reverberation in complex geometries with multiple connected spaces. Such effects are necessary in mixed reality to achieve a sense of presence for virtual people and things within the real world, but have so far been computationally infeasible on mobile devices. We propose a novel cloud-enabled system that enables such immersive audio-visual scenarios on untethered mixed reality devices for the first time.

1 Introduction

Untethered mixed reality presents the unique computational challenge of modeling reality faithfully on a resource-constrained platform, such as the HoloLens or mobile phones. A critical aspect of immersion is interactive sound propagation so that sounds from virtual sources propagate similarly as real sources within their shared real surroundings. For instance, consider a virtual talker walking away from the listener via a doorway into another room. As the talker walks past the door, one would expect their voice to undergo occlusion, smoothly losing loudness due to edge diffraction losses as waves have to bend around the door opening to reach the listener. Further, the talker's speech must be heard as coming from the door ("portaling") rather than directly from their physical location because salient initial wavefronts arrive via the door. Lastly, we expect the speech streaming from another room to have increased reverberance as the emitted sound follows numerous multiply-scattered paths to arrive at the listener through the door.

As a more complex example, consider the listener standing outside a physical room where many virtual talkers are conversing. Without propagation modeling, the listener would hear an implausibly clear, loud soundscape from many directions around them, rather than a faint murmur heard through the door. Such expectations are built into human auditory perception from everyday experience, and plausibly reproducing them within a mixed reality context is critical to preserve immersion. Note that propagation modeling is independent of the quality of binaural or speaker spatialization. The latter serves to faithfully reproduce the sound field around the listener that is interactively output by the propagation system.

Acoustic wave simulation naturally models diffraction effects in complex scenes. However, it has so far been out of the reach of untethered systems because of the fundamental computational challenges, as well as practical issues relating to immersing multiple participants in a shared virtual acoustic space. Our contribution in this paper is to propose a novel synthesis of existing technologies that results in a viable system that, for the first time, allows practical rendering of wave effects discussed above on an untethered mixed reality device: in our tests a HoloLens 2 and a Samsung Galaxy S9 mobile phone.

We demonstrate the system in the accompanying video result. Our system is enabled by two key ideas. Firstly, to utilize recent developments in Mixed Reality systems for visual indexing and search of 3D spaces. Secondly, to offload acoustic simulation to the cloud, where a novel acoustic map persistently associates the space with its acoustic data. As users explore the world, pieces of the acoustic description of the world are populated and downloaded for fast rendering, as needed.

2 Related Work

While there are no prior works on sound propagation for untethered augmented/mixed reality to our knowledge, there is a large body of literature in room acoustics, games and virtual reality.

The Geometric Acoustic (GA) approximation [1] is a commonly used approach for real-time acoustics [2]. GA is derived from the fundamental linear wave equation by taking an infinite-frequency (zero wavelength) limit. This yields the Eikonal equation describing acoustic energy propagation along rays that scatter in the scene to produce reverberation. Stochastic ray tracing has thus been used successfully for modeling diffuse reverberation in room acoustics [3]. In mixed reality scenarios we target, the line of sight between source and listener is often obstructed which makes deterministic diffraction modeling a necessity. Since diffraction is inherently a finite-wavelength effect, GA systems face difficulties in this area, constituting an open problem. A detailed survey of recent techniques including various diffraction approximations is presented by Savioja and Svensson [4].

Most modern GA systems rely on stochastic path tracing (since deterministic path tracing has CPU cost exponential in number of bounces) where one finds many reverberant paths, each with multiple bounces connecting source to listener [5, 6]. A further difficulty is that due to the stochastic nature of the process, it can be hard to know a priori how many paths (each costing CPU) one needs to trace in order to get a reasonably stable estimate. Systematic studies are beginning [6]. Detailed scene shape with numerous triangles worsens performance, so it is common to require the user to provide a simplified geometric model with planar facets. This presents a major challenge for automatically depth-scanned geometry in mixed reality. Automatic acoustic scene simplification is a challenging problem in its own right with limited work [7]. Nevertheless, GA approximations are commonplace - its strength is simplicity of formulation and flexibility for application-dependent modifications. Research systems such as RAVEN [3] allow real-time rendering while requiring the compute power of one or several workstations. This fits the target application of interactive walk-through in computer-aided-design applications.

Mixed-reality systems simultaneously present difficult simulation challenges while offering very limited computational resources. Multi-room spaces are common and the line-of-sight between source and listener is often blocked. Modeling occlusion and portaling in these scenarios requires robust, deterministic diffraction modeling that stays consistent on source/listener motion. Further, with GA, finding complex, multi-bounce paths that connect the source and listener while passing through intervening doors or windows can take a substantially larger amount of stochastic sampling (and hence, CPU) compared to a single-room scene, a well-known problem in computer graphics [8]. Lastly, the scene geometry obtained from depth captures can be noisy, and the system must stay robust to such noise, while ideally not requiring any user intervention for geometry simplification.

Many systems based on GA approximations have been developed for virtual reality but they do not meet the above requirements of untethered mixed reality. Steam Audio [9] employs stochastic path tracing while ignoring diffraction, and requires the user to decide a ray budget for computing acoustic response between source

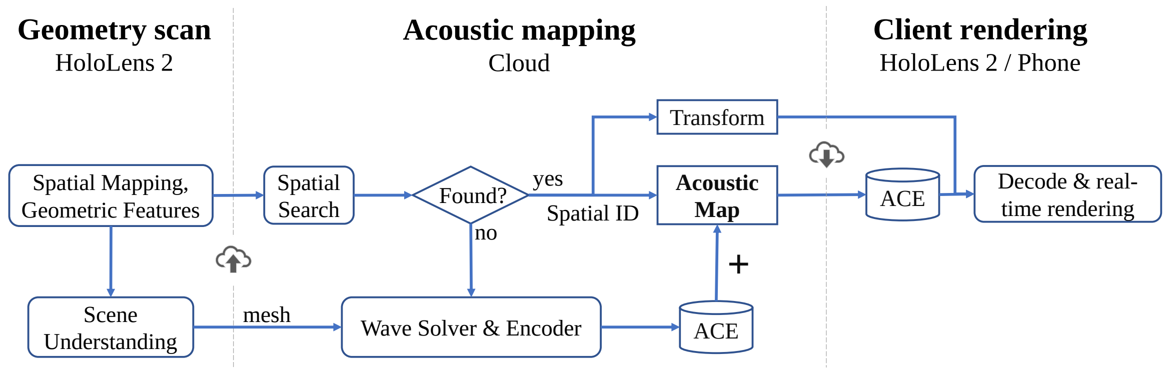

Fig. 1: System architecture

and listener. This transfers the robustness/accuracy versus CPU-cost trade-off to the user. Other software such as Google Resonance [10] are efficient enough to perform audio spatialization processing on mobile devices [11], but propagation modeling is limited: source location is ignored for reverberation modeling and user is required to manually specify key listener locations.

The computational difficulties of diffraction modeling on complex scenes have led to the parallel development of pre-computed wave simulation approaches. These methods were introduced in [12] where volumetric, band-limited wave simulation is performed offline from a set of sampled listener locations and acoustic reciprocity is employed to reduce simulation cost. Precomputation restricts to static scenes but allows the use of first-principles wave simulation directly on complex scene geometry. The resulting acoustic impulse response field is stored within a compressed representation in an acoustics dataset. At runtime, spatial interpolation and frequency extrapolation yield decoded wideband responses for each source that are then applied to its emitted audio. Subsequent work has followed this general architecture.

For the special case of outdoor scenes with few separate scatterers, the equivalent source method was employed in [13]. This results in improved compression but rules out indoor scenes and increases decoding cost substantially, requiring per-frequency multipole summations from each scatterer at the evaluation point. The work in [14] introduced the idea of perceptual compression of acoustic fields. This allows general scenes, fast decoding speed amounting to an interpolated table lookup, and substantially higher compression. Simultaneously, parameterization allows lightweight multi-source rendering methods that avoid per-source convolution. The lightweight system has allowed adoption in major games [15]. Later work in [16] introduced efficient encoding and rendering of directional acoustic effects such as sound re-direction around doors. This has enabled practical usage in VR scenarios [17].

We adopt the implementation of [16] available in the "Project Acoustics" system [18]. The method is attractive for our application because it has a resource-light rendering runtime and the pre-computation can be performed as a massively parallel workload in the cloud since the simulation for each listener position proceeds independently. The resulting acoustics dataset is compact (<10MB), practical for mobile use cases. Wave simulations naturally include diffraction, yielding robust renderings as we show in our results.

3 System Architecture

Figure 1 provides an overview of our system. We briefly discuss each step to provide an overview. Following sections describe each in more detail.

Geometry Scan When the user walks into a space using a mixed reality device, it performs a spatial search using the visual and depth features of the space. If found, a "Spatial ID" is returned, which is a unique identifier for the space. If not, the user is notified and must first scan the physical scene by slowly walking around and pointing the mixed reality device at all the surfaces (i.e. walls, ceiling, floor). The depth-scanned geometry is extracted and then passed through scene completion algorithms to fill in any large missing portions of the walls. In our test implementation, the mixed reality device is a HoloLens 2, and the Azure Spatial Anchors API [19] and Scene Understanding API [20] provide the facilities for spatial search, depth scanning, and scene completion.

Note that we currently only extract geometry information. Accurate acoustic simulations also require per-triangle material information which remains difficult to acquire using visual sensors available on mixed reality devices. Acquiring granular material information is an important avenue for future work. For instance, in lieu of accurate material information, perhaps one could extract the decay time of the room from the user's speech [21] and then use the scene geometry's area and volume in combination with Sabine's equation to determine the scene's average diffuse-incidence absorption coefficient.

Acoustic Mapping If visiting a new space, the triangle mesh representing scene geometry from above is input to a massively parallel acoustic simulation in the cloud. The offline simulation generates a data file containing the acoustic properties of the scene. This data file is added to a persistent cloud acoustic map that relates a location's Spatial ID to the corresponding acoustic data file (shown with "ACE" in Figure 1).

When the user(s) later re-enters the same space, the Spatial Anchor subsystem discovers that it is a known space, returning the Spatial ID which is then used to download relevant data from the acoustic map along with all necessary scene-alignment transforms.

Client Rendering With the appropriate acoustic data and transforms downloaded, the acoustic runtime performs interpolated look-ups appropriately, enabling fast parametric rendering for multiple moving sound sources for each user. This achieves the desired effect of virtual sound sources having comparable sound propagation to physical sound sources located in the real-world space.

4 Geometry Acquisition

Figure 2 shows the theater space used in this study. The chairs represent a practical challenge: they are hard to reconstruct, and indeed the captured triangle mesh looks quite noisy, shown in Figure 3. Our wave simulation approach is robust to such geometric errors, resulting in a graceful degradation in acoustical simulation accuracy. In this instance, the noisy geometry results in more aggressive energy diffusion, but avoids implausible artifacts such as loudness jumps in the rendered audio.

Fig. 2: Theater used in this case study.

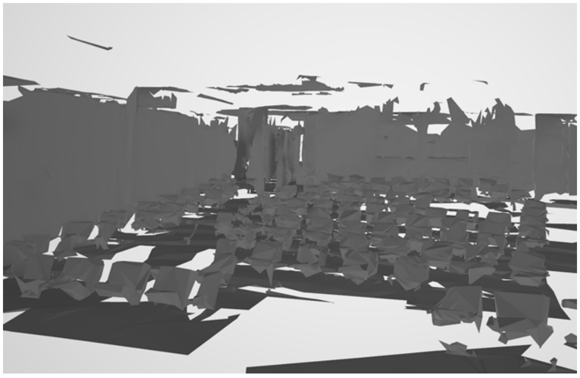

Fig. 3: Depth-scanned geometry of the real-world theater room captured with HoloLens 2.

Notice that while the raw geometry provides a rough outline of the physical room, the geometry itself is often incomplete. With wave simulation, much like reality, small holes cause small perturbations in renderings. This is quite unlike geometric acoustics, where a line-of-sight may be established between source and listener through a small hole causing sudden dis-occlusions. At the same time, holes with large diameter ($\sim$ 1 meter) will cause audible discrepancies in the simulated acoustic properties such as reverberation time and primary arrival direction.

Fortunately, scene completion algorithms [20] can often fix large missing patches for the primary walls in a space. While this process only hallucinates geometry, it does ensure that large amounts of energy leakage is avoided. Figure 4 shows the result of running the theater from Figure 2 through scene completion. To help disambiguate the surfaces, the wall planes have

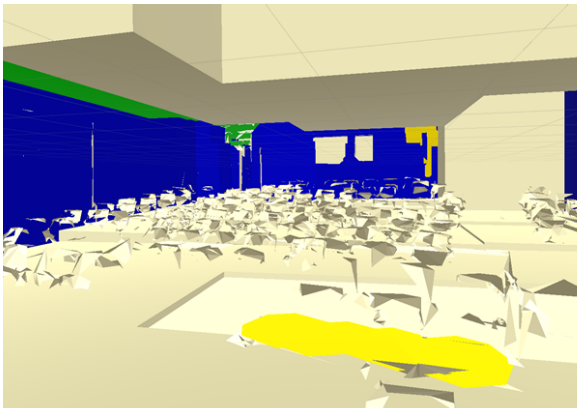

Fig. 4: Cleaned and completed geometry

been colored blue and the platform planes (in this case, a table) have been colored yellow. On occasion, small gaps between surfaces are left so that the meshes are not entirely watertight, but any remaining holes are much fewer and smaller than unprocessed geometry. As discussed above, this constitutes admissible input for acoustic wave simulation.

5 Acoustic simulation and mapping

Following the bottom of Figure 1, the scanned and completed scene geometry from the prior step is passed to the acoustic wave simulator, which first voxelizes the scene geometry and lays out potential listener "probe" locations in the scene [22], shown in Figure 5 in green and cyan respectively. In our current experimental setup, this is a remote PC rather than hosted in the cloud, but that can be easily modified.

Simulation Each listener probe is simulated in a massively parallel fashion on a compute cluster. In our case, the compute cluster was hosted in the cloud. Each simulation proceeds independently without any inter-node communication. Wave simulations are typically band-limited to control compute costs. We empirically found using 500Hz as the maximum simulated frequency offers a good balance of spatial resolution and compute time for the physical spaces in this experiment.

The technique we employ [16] performs frequency extrapolation by rendering parameters computed over the simulated frequency range for the entire audible bandwidth. Horizontal probe spacing was limited to

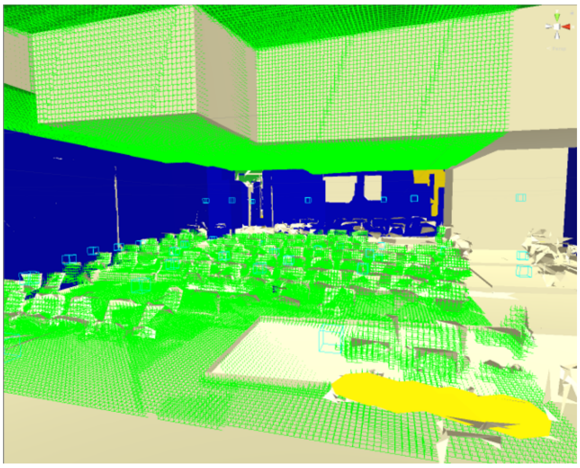

Fig. 5: Scene voxelization employed by wave solver

a maximum of 1.5 meters to control interpolation errors. In tight spaces, the spacing reduces automatically to ensure narrow hallways aren't missed by sampling [22]. Larger horizontal spacing values were observed to produce inaccuracies in the primary arrival direction for sources very close to the listener.

Acoustic Data The overall acoustic data for the scene consists of a concatenation of per-listener-probe volumetric parametric data that encodes the acoustic response for a point source moving in 3D for a listener fixed at the probe location. The encoding consists of a set of perceptual parameter fields as described in [16]. This is depicted with the "ACE" file in Figure 1. For the theater space we test, the size of this file is 2.5MB.

Mapping Upon completion of processing, the resulting data file is saved in a persistent "Acoustics Map" in cloud storage which tabulates the 3-tuples (Spatial ID, Transform, ACE) indexed by the Spatial ID of the space, as generated by the Spatial Anchors subsystem. Following the top of Figure 1 from left to right, when a user visits the same theater space, the Spatial Anchor query based on geometric features succeeds - the user need not do detailed geometry scanning this time - returning the Spatial ID of the recognized space. This is used as the key for looking up associated acoustic data and a transform that tells the current user the pose of the scene when acoustic processing was performed.

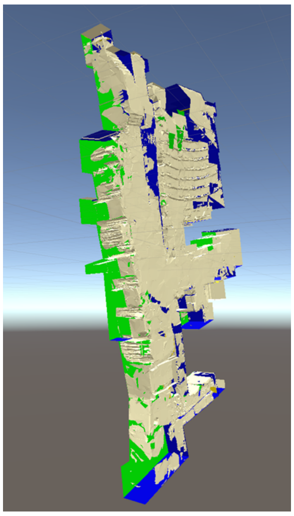

Fig. 6: An example scene pose without using transform alignment via Spatial Anchors.

6 Client rendering

The identified transform and acoustic data is downloaded to the user's device and can then be used to auralize arbitrary moving virtual sources in the scene over headphones or a mixed reality device's built-in speakers.

Aligning transforms In virtual reality, it is easy to align various objects because there is only a single reference coordinate system, namely that of the simulation engine. A perhaps surprising practical complexity in mixed reality is this single origin concept no longer applies, requiring a transform alignment scheme (see Fig. 6). To realize proper alignment between the virtual and physical worlds, Spatial Anchors are used to define shared, persistent coordinate systems. A Spatial Anchor is world-locked in physical space and when backed with a cloud-enabled Spatial Anchor subsystem, any device visiting a space can access the same Spatial Anchor from the cloud using the Spatial ID. Along with this single, shared, coordinate system, one must also transform between coordinate system definitions for the various components, which we list below for completeness.

- *Scanned Geometry*: right-handed, Y-up
- *Unity*: left-handed, Y-up
- *Acoustics Engine*: right-handed, Z-up

In all cases, +X points left to right on screen. With all transforms aligned, multiple devices are able to immerse in a common virtual acoustic space along with physical sound sources.

Rendering Each device performs decoding of the acoustic data via interpolated look-ups in parameter fields for each source at an update rate of 47 Hz (1024 sample audio buffer size at 48kHz sample rate). Because the acoustics simulation was done offline, all that is required at runtime is a table lookup which takes around $10\mu s$ to complete per source. This results in a set of perceptual acoustic parameters for the source which are translated into a lightweight and scalable parametric signal processing pipeline [16]. The translation process also allows on-the-fly design modifications [17] such as for dynamically tuning the T_{60} decay time to compensate for incorrect material assignments. We did not need to perform such modifications, and instead used an energy absorption coefficient of 0.1 for all surfaces in our simulations. As mentioned previously, material detection is an area for future improvement.

The rendering process uses Head-Related Transfer Function (HRTF) spatialization of the direct sound based on decoded arrival direction, distance attenuation, and low-pass filtering based on diffraction loss. Reverberation is rendered by weighted sends to a *fixed* set of three colorless convolutional reverberation filters with varying decay times. This scheme, first presented in [14], approximately renders the decoded reflections loudness and decay time for each source while avoiding instantiating a reverberation filter per sound source, which would be prohibitively expensive on a mobile device. The rendering is thus scalable: it can handle up to 30 sources in real-time on a Samsung Galaxy S9 and a HoloLens 2 in our tests. In the latter case, we leverage the spatialization hardware-offload feature. This completely removes all HRTF processing from the CPU, leaving it only to handle the reverberation filters.

7 Implementation

For our implementation, the mixed reality device was the HoloLens 2. The HoloLens 2 is capable of both tracking and mapping the environment's geometry using its on-board sensors and processing. While we use the HoloLens 2 for geometry acquisition in our case

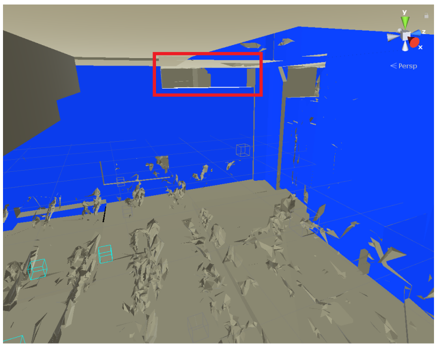

Fig. 7: A hole in the capture geometry, marked with red rectangle.

study, increasingly such scene reconstruction capabilities are available on mobile phones as well [23].

The Scene Understanding API [20] available on HoloLens 2 uses semantic completion to fill-in the walls and floor with flat planes. These planes are disambiguated with proper labels. It will also pick out surfaces such as tables.

Unity game engine was used for audio-visual rendering with the Project Acoustics plugin [18] providing the acoustic wave simulation engine. Project Acoustics uses Azure Batch [24] to provide cloud resources as a virtual computer cluster. Each simulation uses the ARD wave solver [25], followed by an encoding step as described in [16].

Azure Spatial Anchors[19] is used to serve as the cloud-enabled Spatial Anchor subsystem. Azure Spatial Anchors provides a service for persisting and identifying Spatial Anchors across devices.

8 Results

Our primary result is a demonstration of the working system: `https://aka.ms/AA8a8ca`. Please consult this video as you read the descriptions below. The demo contains live footage of our application running on a HoloLens 2 in the theater space. This footage was obtained using capture technology [26] that yields an authentic first-person account of our system as seen and heard with the HoloLens 2.

We start with a screenshot and wave simulation on the space. While the simulation is in full 3D, showing a 2D slice makes it easier to visualize the result. Acoustic pressure fluctuations are color-coded (red positive, blue negative). One can see acoustic wave-fronts moving through the space and diffracting around an open doorway at bottom right. The geometry is not perfectly water-tight and the simulation is quite resilient to that, except substantial leakage into the hallway occurring at the top right of the theater room which is due to a window-sized hole in the scanned geometry shown in Figure 7. This could be fixed with an improved scan by the user. We used the result as-is to illustrate practical issues faced.

We then show a debug view of an audio emitter moving through the theater. The mesh input into the acoustics simulation is overlaid on the real world. From this view, it is apparent that there are many inaccuracies in the captured geometry, especially the chairs, and the ceiling is at the wrong height in some places. However, these inaccuracies did not result in a jarring implausibility in the experience, at least in this initial study.

We then show two A-B comparisons first with only HRTF processing and then with full acoustic processing. The first comparison demonstrates the in-room reverberation modeling provided by the acoustics simulation. The second comparison demonstrates occlusion, portaling, and reverberation changes as the listener moves further out of the room. Notice how acoustics update smoothly as the listener walks around the space, with the source's arrival direction adjusting appropriately to the doorway, correctly reinforcing the visual presence of a door. Such effects are critical for mixed reality immersion in scenes that often go beyond a single enclosure.

In Figure 8, we show horizontal slices of a few important parameter fields for our test scene at two listener locations demonstrated in this last clip, one inside the theater (top) and another just outside the theater (bottom). The complete set of parameters is described in [16]. Looking at the top left, the "Direct Loudness" field encodes diffraction losses on the initial wavefronts propagating from source to listener. Distance attenuation is factored out so that free-space propagation results in a constant 0dB field. For any source inside the theater, the initial (dry) audio has little diffraction loss for propagation to the listener inside the theater but a source outside the entrance will have significant occlusion. Therefore, the main hall is yellow, close to 0dB, while the losses progressively increase as the source leaves through the entrance to the hallway outside. When the listener is outside the theater (bottom

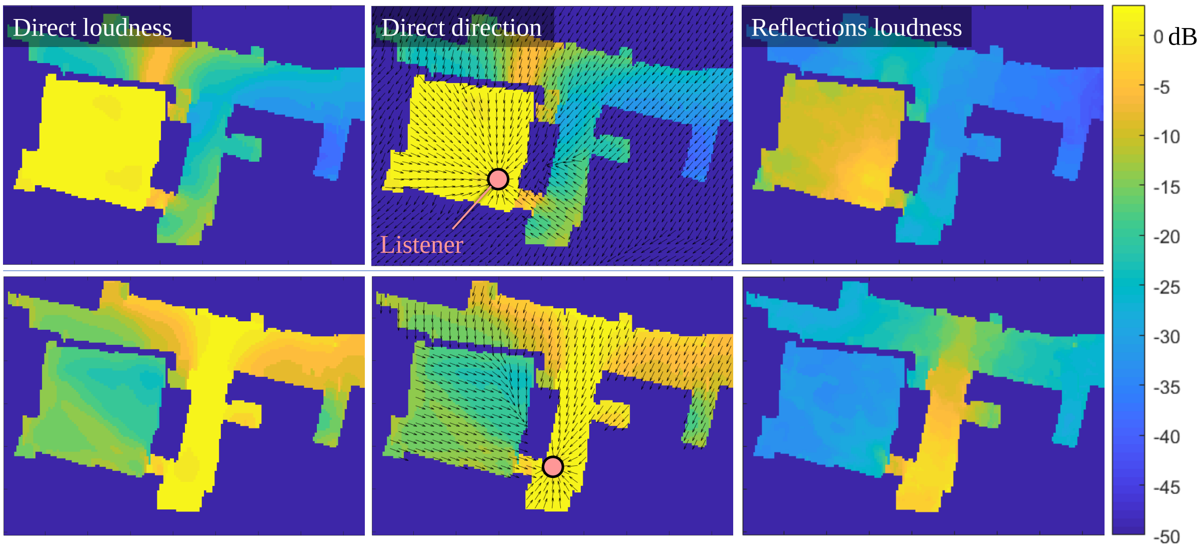

Fig. 8: Summarized acoustic data for listener standing inside (top row) and outside (bottom row) the theater. Heat maps show parameter values for source location varying over the image. The complete dataset for the scene contains similar data stored for many potential listener ("probe") locations shown with blue boxes in Fig. 5.

left) the opposite happens: sources in the hallway outside the theater are loud, those inside are faint. Similar trends can be observed in the loudness of reflections, shown in the right column.

The middle column shows the arrival direction of spatialized audio for various source locations. For instance, in the bottom middle figure, most arrows inside the theater point from the entrance at the bottom to the listener; for any source at these locations inside the theater, the sound is heard at the listener (pink circle) as coming from the open door. Some arrows in the top-right corner of the theater point downwards, because sound leaks from the hole discussed previously (Figure 7) to arrive at the listener via hallway. With a wave solver, errors in geometry translate to physically intuitive changes. In fact, our rendered results could be used to guide a user to find the hole to improve the geometry scan. One would place a virtual source outside the theater and then a user inside the theater can just walk around listening for where the sound is coming from, to locate the hole.

9 Conclusion and Discussion

We showed a proof-of-concept of a novel cloud-enabled system that for the first time enables interactive sound propagation with immersive wave effects in untethered mixed reality. We were able to generate acoustic data in the cloud from real-world geometry captured from a HoloLens 2, persist it in an acoustic map, and share the data across any new device in the space. Combined with lightweight parametric rendering, we are able to render immersive acoustic effects of virtual sources in the real world.

Much remains for the future. We currently do not detect material properties of the scanned geometry which are important for acoustical fidelity. The acoustics data was verified to be well-aligned with the listener within 5-10 meters of the Spatial Anchor origin, but when traveling further away the alignment error increases in proportion to the distance away from the Spatial Anchor ("lever-arm effect"). Improved alignment techniques are needed in the future. We also wish to perform controlled side-by-side comparisons of real vs. virtual sounds.

Despite limitations, we believe the system as demonstrated is already quite promising for many mixed reality applications where a plausible rendering is sufficient. It could also prove a useful research platform for ecologically-valid perceptual evaluations into the required degree of geometric and acoustical fidelity for convincing mixed reality experiences. Such studies would help inform further research in the field of mixed reality audio.

References

[1] Pierce, A. D., *Acoustics: An Introduction to Its Physical Principles and Applications*, Acoustical Society of America, 1989, ISBN 0883186128.

[2] Vorländer, M., *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality (RWTHedition)*, Springer, 1 edition, 2007, ISBN 3540488294.

[3] Schröder, D. and Vorländer, M., "RAVEN: A Real-Time Framework for the Auralization of Interactive Virtual Environments," in *Forum Acusticum*, 2011.

[4] Savioja, L. and Svensson, U. P., "Overview of geometrical room acoustic modeling techniques," *J. of the Acoustical Soc. of Am.*, 138(2), pp. 708–730, 2015.

[5] Schröder, D., *Physically Based Real-Time Auralization of Interactive Virtual Environments*, Logos Verlag, 2011, ISBN 3832530312.

[6] Cao, C., Ren, Z., Schissler, C., Manocha, D., and Zhou, K., "Interactive Sound Propagation with Bidirectional Path Tracing," *ACM Transactions on Graphics (SIGGRAPH Asia 2016)*, 2016.

[7] Siltanen, S., *Geometry Reduction in Room Acoustics Modeling*, Master's thesis, Helsinki University of Technology, 2005.

[8] Veach, E. and Guibas, L. J., "Metropolis Light Transport," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, p. 65–76, ACM Press/Addison-Wesley Publishing Co., USA, 1997, ISBN 0897918967, doi: 10.1145/258734.258775.

[9] Valve, "Steam Audio," `https://valvesoftware.github.io/steam-audio/`, 2018, accessed Nov 2018.

[10] Google Inc., "Resonance Audio," `https://developers.google.com/resonance-audio/`, 2018, accessed Nov 2018.

[11] Gorzel, M., Allen, A., Kelly, I., Kammerl, J., Gungormusler, A., Yeh, H., and Boland, F., "Efficient Encoding and Decoding of Binaural Sound with Resonance Audio," in *AES International Conference on Immersive and Interactive Audio*, 2019.

[12] Raghuvanshi, N., Snyder, J., Mehra, R., Lin, M. C., and Govindaraju, N. K., "Precomputed Wave Simulation for Real-Time Sound Propagation of Dynamic Sources in Complex Scenes," *ACM Transactions on Graphics*, 29(3), 2010.

[13] Mehra, R., Raghuvanshi, N., Antani, L., Chandak, A., Curtis, S., and Manocha, D., "Wave-based Sound Propagation in Large Open Scenes Using an Equivalent Source Formulation," *ACM Trans. Graph.*, 32(2), 2013, ISSN 0730-0301, doi:10.1145/2451236.2451245.

[14] Raghuvanshi, N. and Snyder, J., "Parametric Wave Field Coding for Precomputed Sound Propagation," *ACM Trans. Graph.*, 33(4), 2014, ISSN 0730-0301, doi:10.1145/2601097.2601184.

[15] Raghuvanshi, N., Tennant, J., and Snyder, J., "Triton: Practical pre-computed sound propagation for games and virtual reality," *J. Acoustical Soc. of Am.*, 141(5), pp. 3455–3455, 2017.

[16] Raghuvanshi, N. and Snyder, J., "Parametric directional coding for precomputed sound propagation," *ACM Trans. on Graphics*, 2018.

[17] Godin, K. W., Rohrer, R., Snyder, J., and Raghuvanshi, N., "Wave Acoustics in a Mixed Reality Shell," in *2018 AES Intl. Conf. on Audio for Virt. and Augmented Reality*, 2018.

[18] Microsoft Corp., "Project Acoustics," `https://aka.ms/acoustics`, 2018, accessed Jan 2019.

[19] Microsoft Corp., "Azure Spatial Anchors," `https://azure.microsoft.com/en-us/services/spatial-anchors/`, 2019, accessed Dec 2019.

[20] Microsoft Corp., "Scene Understanding," `https://docs.microsoft.com/en-us/windows/mixed-reality/scene-understanding`, 2020, accessed March 2020.

[21] Gamper, H. and Tashev, I., "Blind reverberation time estimation using a convolutional neural network," in *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 1–5, IEEE, 2018, nominated for best paper award.

[22] Chaitanya, C. R. A., Snyder, J., Godin, K., Nowrouzezahrai, D., and Raghuvanshi, N., "Adaptive Sampling For Sound Propagation," *IEEE Transactions on Visualization and Computer Graphics*, 25(5), pp. 1846–1854, 2019.

[23] Google Corp., "Google ARCore," `https://developers.google.com/ar`, 2020, accessed Apr 2020.

[24] Microsoft Corp., "Azure Batch," `https://azure.microsoft.com/en-us/services/batch/`, 2019, accessed Dec 2019.

[25] Raghuvanshi, N., Narain, R., and Lin, M. C., "Efficient and Accurate Sound Propagation Using Adaptive Rectangular Decomposition," *IEEE Transactions on Visualization and Computer Graphics*, 15(5), pp. 789–801, 2009, ISSN 1077-2626, doi:10.1109/tvcg.2009.28.

[26] Microsoft Corp., "Mixed Reality Capture," https://docs.microsoft.com/en-us/hololens/holographic-photos-and-videos, 2020, accessed April 2020.