VIRTUALIZED ACOUSTICAL MODELING, SIMULATION, AND AURALIZATION OF A 3D ENVIRONMENT USING SONEL MAPPING

Andrei LĂPUȘTEANU[1], Anca MORAR[2], Alin MOLDOVEANU[3], Florica MOLDOVEANU[4], Maria-Anca BĂLUȚOIU[5], Victor ASAVEI[6]

Virtualized acoustical simulations, alongside auralization techniques, can provide valuable insight regarding the acoustical characteristics of 3D environments and have already seen usage in architectural design software. The proposed pipeline builds upon the Sonel Mapping method a stochastic particle-based energy propagation framework. We extend and improve its design by providing an intuitive tool for prototyping simple 3D simulation environments, employing a voxel-based acceleration algorithm, as well as an auralization method which is coupled with a novel approach for estimating sound directionality even in the absence of a direct acoustic path between source and receiver.

Keywords: acoustical simulation, auralization, Sonel Mapping, binaural room impulse response, HRTF

1. Introduction

Virtualized acoustical simulations aim to obtain the reverberant properties of an environment as a result of algorithms which approximate the propagation of acoustical waves. Such methods have been used in a variety of applications, the majority of which are tailored to the needs of acoustical engineers which work in the field of architectural acoustics. Acoustical simulations analytically convey sound characteristics of a virtual scene, which can be tweaked to yield the desired results. If sufficiently accurate, these can be used in the design of real

[1] PhD Student, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: andrei.lapusteanu@upb.ro
[2] Assoc. Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: anca.morar@cs.pub.ro
[3] Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: alin.moldoveanu@cs.pub.ro
[4] Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: florica.moldoveanu@cs.pub.ro
[5] PhD Student, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: maria_anca.balutoiu@upb.ro
[6] Assist. Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: victor.asavei@cs.pub.ro

environments. Also, through a process called auralization [1], the results of the above-mentioned simulations can be acoustically rendered, thus offering a perceptual validation of the acoustical characteristics. The development of a software capable of accurately simulating the sound properties of an environment can prove to be challenging due to the complexity of physical phenomena which require computation. After settling on a simulation model, the environment's geometrical description needs to be defined. Also, specific acoustic material properties, which dictate the result of sound interaction, should be specified for each surface. Depending on the simulators' use case, a plethora of acoustical parameters can be computed, some of which may be required to perform the auralization process.

2. Related work

Acoustical simulation methods. The studied acoustical simulation methods can be broadly grouped into 2 categories [2]: wave-based and geometric methods. Wave-based simulations aim to solve the wave equation, also known as the Helmholtz-Kirchhoff equation [3], which describes how waves propagate through a medium. Wave-based simulations are advantageous because they can intrinsically model phenomena such as diffraction and interference [4] (especially at low frequencies, where the wavelength is comparable in size with the scene geometry), and they generally yield accurate results [5]. These simulations require a significant amount of time to complete, a fact that renders them generally unsuitable for real-time applications [2]. The most used approaches for wave-based simulations are Finite Element Method (FEM) [6], Boundary Element Method (BEM) [7], and Finite-Difference Time-Domain Method (FDTD) [8]. The FDTD approach has seen plentiful research, as it runs directly in the time domain, benefiting from a simpler implementation. The work described in [9] uses a method called Adaptive Rectangular Decomposition to accelerate computations, and papers [10] [11] employ GPU-accelerated solvers. Mehra et al. present in [11] [12] [13] the Equivalent Source Method for real-time outdoor simulations, and a method which aims to implement wave propagation in games can be studied in [14].

Geometrical simulation methods compute sound interactions by modeling sound as rays [2], similar to methods used for computing global illumination in computer graphics. These so-called acoustical rays are traced through the environment and their interactions are computed. Mathematical models are used to simulate propagation effects – such as absorption caused by distance and atmospheric effects – as well as the result of interactions with surfaces, which differ based on the acoustical materials of the environment's geometry. In contrast with wave-based methods, geometrical simulation methods are generally easier to

implement and offer significantly lower computation times, therefore being more suitable for real-time applications. The nature of the acoustical modeling of geometric methods prevents them from accurately simulating phenomena which occur at lower frequency, thus rendering them less accurate than wave-based methods [15]. Plenty of work has been carried out for geometric simulation methods, yielding a plethora of different approaches, which can be split into deterministic techniques, which produce the same result after every simulation, and stochastic techniques, which make use of random sampling to approximate the simulated sound field [5]. Deterministic methods include the Image Source method, with notable work carried out in [16] [17] [18], and the Beam Tracing method, for which a number of implementations and improvements can be studied in [19] [20] [21]. Stochastic methods include Ray Tracing, which has seen plentiful usage in acoustic simulators, starting with Krodstad et al. [22], who were the first to propose computing impulse response using Ray Tracing. More recently, works described in papers [23] [24] [25] propose methods to resolve the intrinsic shortcomings of geometrical methods. There's also Particle Tracing, which has methods such as Phonon Tracing [26] and Sonel Mapping [27]. Among the stochastic methods we also mention Frustum Tracing, with various implementations and improvements [28] [29] [30].

Commercial solutions. A number of freely available frameworks can be used to perform acoustical simulations. NVIDIA WRWorks Audio [31] is a framework which uses path-tracing to simulate and render sound in real-time and can handle sound reflection, absorption, transmission, and even approximate diffraction effects, as well as filter the sound using Head-Related Transfer Functions (HRTFs). Steam Audio [32] offers a 3D sound API tailor-made for game development in Unity and Unreal Engine 4. Wayverb [15] is an open-source acoustics simulation software which aims to solve the issues of both wave and geometry-based methods by creating a hybrid model which implements the Image Source and Ray Tracing geometric methods for higher frequencies, as well as the Waveguide method for lower frequencies.

Sonel Mapping. As the method proposed in this paper is based on Sonel Mapping, a short overview of its methodology is presented for the remainder of this section. Sonel Mapping was developed by Kapralos et al. [27] and is a geometrical-based stochastic simulation method, capable of modeling specular and diffuse reflections, as well as absorption and diffraction effects. The method was inspired by Photon Mapping [33], which is a two-stage technique used to compute global illumination. A sonel (or sound element) is defined as a packet of information, which propagates from source to receiver, and contains the information needed to perform the acoustic simulation, such as the position in space, sound frequency, and travelled distance. Its stochastic nature is determined by a Russian Roulette technique [2], which probabilistically determines the result

of ray-surface interactions. Sonel Mapping is comprised of 2 stages, namely Sonel Tracing and Acoustical Rendering.

The purpose of the Sonel Tracing stage is to populate a map of sonels. Rays are emitted and traced from the sound source, and all interactions with surfaces are computed. The result of the interaction is determined probabilistically. A uniformly distributed random number is generated, and with regard to the acoustical material assigned to the surface, a possible outcome is selected. If the ray does not lie within the diffraction zone, it can be either terminated in case of absorption, reflected specularly or diffusely, in which case the tracing of the ray continues. Only for diffuse reflections the sonel is added to a k-d tree, called a sonel map.

The Acoustical Rendering stage uses the previously computed sonel map in order to estimate the frequency dependent echograms, which represent the temporal distribution of the energy of the sound waves that arrive at the receiver. Visibility rays which interact with the surfaces of the environment are emitted from the receiver. Similar to the Sonel Tracing stage, an interaction outcome is computed probabilistically, and in the case of diffuse reflections, the sonel map is used to estimate the acoustical energy at the ray-surface intersection. After attenuating and determining the total time of the acoustic path, the computed sound energy is added to the echogram. Specular reflections and absorptions are handled in a manner similar to the previous stage.

3. Theoretical aspects

Sound is a mechanical wave which requires a transmission medium, is characterized by areas of low and high pressure which oscillate about a fixed point in order to transfer energy. Sound waves can be rather complex and aperiodic, sound tones being composed of a superposition of multiple sinusoidal functions **[27]**. The more physical phenomena an acoustical simulator can model, the more accurate the results of a simulation can be. Such phenomena include:

- Reflection (specular and diffuse);
- Absorption;
- Diffraction;
- Refraction or transmission (optional);
- Interference (optional).

Refraction and interference were not modeled in the majority of studied simulators as their effects can usually be considered negligible [2].

Reverberation. An important characteristic in acoustical simulations is reverberation, which can be described as the multitude of waves that propagate through and interact with the environment, eventually arriving at the receiver at different moments in time (due to reflections) [2]. As it can be observed in Fig. 1

(left), sound that is delayed due to reflections is also attenuated. Another relevant characteristic used in simulators is the reverberation time, $\mathbf{RT_{60}}$, which measures the time required by the sound pressure level (SPL) to decrease by 60dB [34].

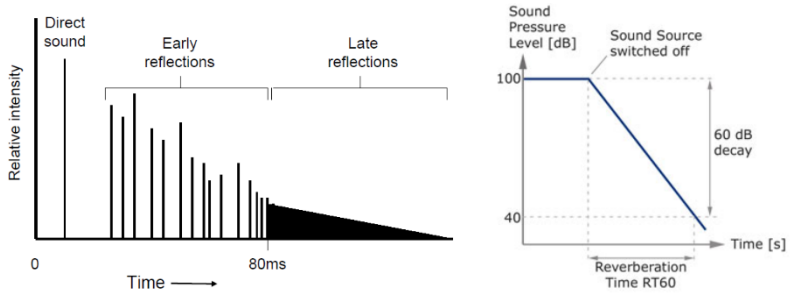

Fig. 1. Reverberation graph (left, source: [2]) and reverberation time (right, source: [35])

Sound versus light. Several assumptions need to be put into place when simulating sound, especially when implementing ray-based methods, which typically find usage in computing illumination in computer graphics techniques. Most notably, the propagation velocity of sound needs to be considered, as its impact is significant in sound simulations. Furthermore, attenuation needs to be taken into account – in comparison, the attenuation of light as a result of the transmission medium can usually be ignored [2].

Acoustical materials and frequency bands. The make-up of physical materials dictates the way in which they interact with sound waves. The thickness of the material can determine sound transmission effects, surface roughness can impact the type of reflection, be it specular or diffuse, whilst porous materials, such as fabrics, are capable of absorbing a large portion of the sound wave's energy [36]. Also, these materials exhibit properties which depend on the frequency of the incident wave. As such, it is common practice to split the audible spectra into multiple frequency bands, and to assign different parameter values for each one.

Echograms and room impulse response (RIR). In the case of geometric-based methods, simulation results can be represented graphically as the temporal distribution of the sound waves' energy that arrives at the receiver [37]. This representation bears different names, such as echogram, reflectogram, energetic impulse response or energy decay curve. These results cannot be directly used to compute the reverberated sound as they only represent the energy arriving at the receiver at different moments in time and thus need to be converted into an equivalent pressure signal, which is in fact the room impulse response. This represents the measured sound pressure as a function of time at the receiver's position [38]. Physically, it is constructed by measuring the result of emitting a very short acoustic pulse (a theoretical idealization would be the generalized Dirac function, and a real-life analogy, a hand clap), but for virtual applications, the RIR

needs to be computed. Several methods of converting the echogram into a RIR have been identified in literature. Kuttruff presents in [39] a method for computing a RIR based on Fourier transforms and a statistically constructed phase spectrum. He refines the method in [40], where he uses the Hilbert transform in order to derive the phase spectrum of the signal with a specially constructed signal configuration from the echogram samples. Dalenbäck presents a method [41] also based on the Hilbert transform, in which he separates the early and late parts of the echogram. The method used in this paper follows the techniques proposed in [37], in which the RIR can be computed in several basic steps:

- Reconstruct the pressure data from each echogram
- Filter and combine the individual responses into a single RIR
- Apply post-processing filters to the RIR

Sound localization. Head-related transfer functions (HRTF). The Duplex theory [42] was the first to try and describe sound localization using interaural time delay (ITD) and interaural level delay (ILD). Batteau's research [43] built upon the theory by including the physical make-up of the receiver into the theory (the pinna, head, shoulders, and upper torso). All of these factors can be collectively represented as a set of filters, known as head-related transfer functions, and encompass information regarding sound directionality, ITD, ILD, as well as differences in spectrum caused by sound propagation. HRTFs are constructed by placing specialized in-the-ear or behind-the-ear microphones on human participants or mannequins and recording sounds emitted at known angles and distances relative to the head. These recordings are further processed to obtain the transfer functions. Datasets, such as the ARI HRTF Database [44], offer a large number of recordings which can be used in applications.

Binaural room impulse response (BRIR). The binaural room impulse response can be regarded as the acoustic "signature" of a room for a particular source and receiver configuration [2]. It is computed by combining the RIR of a particular environment and the appropriate HRTF filters. As such, the BRIR will encompass both reverberation and sound directionality information. The convolution of the BRIR with an anechoic signal (audio signal without echo) will generate the audio used in auralizations [45].

Auralization. It represents the process of creating audible sound as a result of acoustically simulating a virtual environment. Vorländer describes auralization as "the technique of creation and reproduction of sound on the basis of computer data" [1]. Whilst acoustical simulations convey plenty of information regarding the acoustical characteristics of an environment, the data has to be analysed and interpreted by specialists. Auralization can take these results a step further and provide a perceptual validation of the environment's acoustics. Methods which propose the integration of auralization into architectural design pipelines have already been defined, as paper [46] showcases, especially in the

design of enclosures for which acoustics play a significant role (concert halls, theaters). Furthermore**,** real-time auralization solutions have been implemented and used in virtual reality applications [47].

4. Proposed workflow

The currently implemented software pipeline can be broadly split into several stages, namely scene description, scene voxelization, acoustical simulation, and auralization. All of these stages have been implemented in the Unity engine on the CPU. Fig. 2 provides an overview of the proposed workflow.

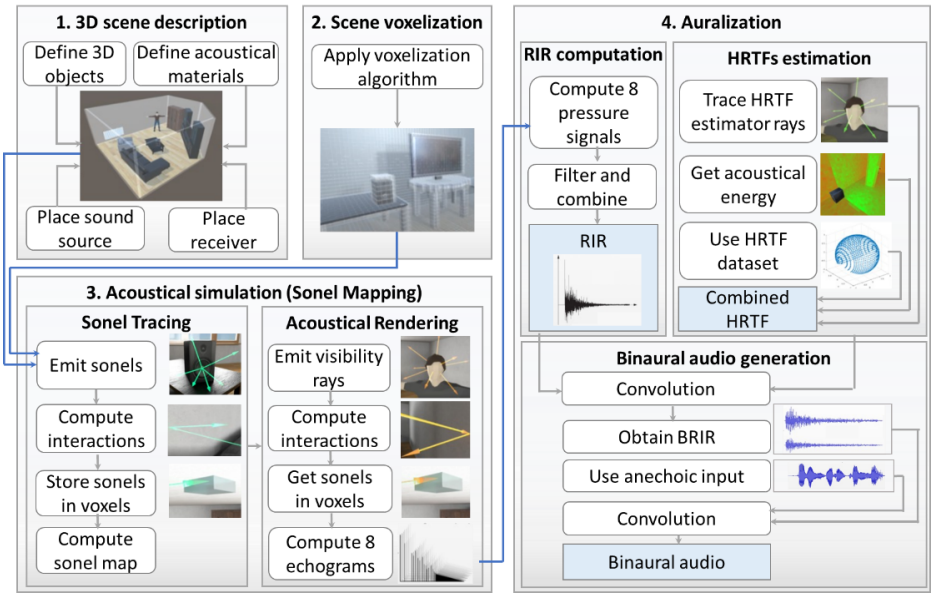

Fig. 2: Proposed workflow

Scene description. In order to perform the acoustical simulation, the virtual objects of the scene, alongside their parameters, need to be specified. A number of 3D models (representing common household furniture) were created specifically for this project. Since objects are made of different materials, the meshes of the 3D objects were designed in order to properly separate these regions. In Unity, each mesh was then assigned a different acoustical material.

Fig. 3. Designed models (left), example of acoustical materials assigned to a 3D object (right)

As the current solution is only suitable for room acoustics, a module capable of defining bounded volumes (air-tight spaces) was implemented. The user is able to easily define rooms with walls placed at arbitrary angles, and these enclosures are automatically detected. This was made possible using graph theory, where each wall is represented as an edge, and wall intersections and endings are abstracted as nodes. Bounded volumes are marked as such by identifying all chordless cycles [48] in the resulting wall graph.

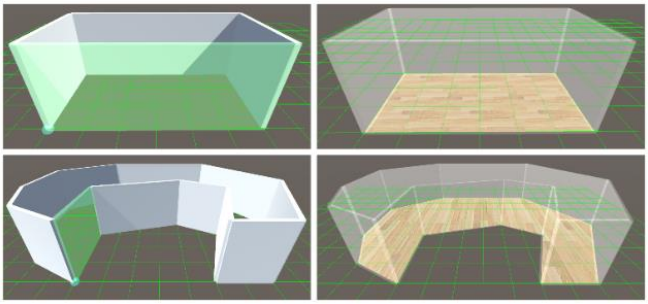

Fig. 4. Example of arbitrary room configurations. The left column depicts the building mechanism, before placing the last wall (green), the right column illustrates the identified bounded volumes

Scene voxelization. In order to accelerate the acoustical simulation, a voxelization algorithm was developed that recursively subdivides the entire scene into voxels. These voxels are not instantiated in the scene, but their positions are stored inside a 3D matrix which is contained only in memory. During the Sonel Tracing stage of the implemented acoustical simulator, if a ray intersects a surface diffusely, the sonel is stored in the voxel containing the point in space of the interaction. In the original Sonel Mapping method, in the Acoustical Rendering stage, when a ray interacts with a surface, the k-d tree needs to be accessed in order to identify the sonels closest to the interaction point. Using this voxelized structure, only a simple mapping from a point in space to the 3D matrix indices is performed. The appropriate voxel is identified, and the sonels contained inside it can be extracted, alleviating the computation needed to identify the closest sonels from the k-d tree.

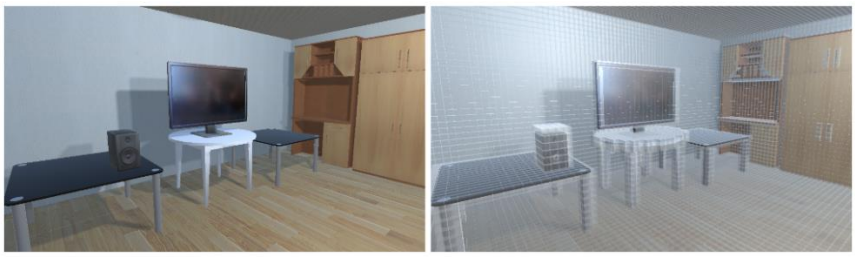

Fig. 5. Scene before (left) and after (right) voxelization. As voxels are not instantiated in the scene, the right image only depicts their supposed configuration

Acoustical simulation. In this stage the Sonel Mapping method is employed to compute the acoustical simulation. The output of the algorithm is represented by the set of accumulated echograms for a particular source and receiver configuration. The current implementation closely follows the methods proposed in the original Sonel Mapping paper – small adjustments were made as needed during development. Currently, the effects of absorption, as well as specular and diffuse reflections are taken into account.

Sonel Tracing. The sonel emitter is modeled as a speaker from which sonel rays are traced through the scene. In order to refine the results at run-time, a number of parameters such as the number of rays, the rays' bounce count, the speaker's cone angle and the sound power level can be modified. Specular reflections are easily computed using Unity's built-in methods, and the direction for diffuse reflections, ω_d, described in spherical coordinates, is modeled using the following equation [2]:

$$\omega_d = (\theta, \phi) = cos^{-1}\left(2\pi\xi_2, \sqrt{\xi_1}\right) \tag{1}$$

where Φ is the angle with the surface normal, θ represents the rotation around the normal and $\xi_1 \in [0 \ldots 1]$, $\xi_2 \in [0 \ldots 1]$ are uniformly distributed random numbers.

For each diffuse reflection, the appropriate voxel for the point in space of the interaction is identified, the sonel is stored inside it, and a new ray is emitted using the previously computed direction, ω_d. In case of absorption, the tracing of the ray is terminated.

Acoustical Rendering. In this stage, so-called visibility rays are emitted from the receiver's position and are traced through the scene. Analogous to the previous stage, a number of parameters can be adjusted at run-time and the echograms' total time and temporal spacing can be defined. Specular reflections and absorption phenomena are computed as described in the Sonel Tracing stage. In the case of diffuse reflections, the tracing of the visibility ray is terminated, and the voxel corresponding to the point in space of the interaction is identified. For each sonel contained in the voxel the total time of the acoustic path is computed and the energy is attenuated with regard to the travelled distance and the atmospheric conditions. The attenuation due to distance is computed as such [49]:

$$L_p = L_W - \left|10\log\left(\frac{Q}{4\pi * r^2}\right)\right|, \tag{2}$$

where L_p [dB] is the sound power level (SWL), representing the energy which can be transmitted continuously by the sound source and is independent of distance. L_w [dB] is the sound pressure level (SPL), which represents the energy that can be transmitted continuously by the sound source and is dependant on distance. Q is the directionality factor and r [m] represents the distance.

The mathematical apparatus used to calculate the attenuation due to atmospheric factors can be found at [50]. It takes into account variables such as air temperature, relative humidity, and pressure. Also, in order to compute the propagation time of an acoustic path, the sound speed needs to be determined. The model described in [51] was implemented – this takes into account air temperature as well as several other atmospheric properties in order to provide an accurate result.

Finally, using the attenuated energy of the sonel and the propagation time, an entry to the corresponding bin of the data structure representing the echogram is added. The process continues for each visibility ray and for each frequency band, and at the end of the Acoustical Rendering stage, each echogram should contain sufficient data in order to perform further computations.

Auralization. In this stage we made use of several signal processing algorithms in order to define our auralization technique. The result of these methods is represented by the binaural audio.

Computing the frequency dependent RIRs. The echograms, which represent the temporal distribution of the energy of the sound waves that arrive at the receiver, need to be converted into a pressure signal which represents the RIR. The implemented method follows the steps described in [37]. Firstly, a windowing function f_δ, defined on the length δ of the echogram, is constructed:

$$f_\delta(x) = 1 - \sin\left(\frac{2\pi x}{\delta}\right) \tag{3}$$

Secondly, using the aforementioned windowing function, a pressure-time function $P(t)$ is constructed for each frequency band by taking the integral over time of the square root of each echogram's energy:

$$P(t) = \sum_i \frac{\sqrt{(E_i)}}{\delta_i} f_{\delta_i}(t - \overline{t_i}), \tag{4}$$

where E_i is the echogram value at the current time interval, or echogram bin, i. t represents the total time (with respect to the sampling frequency) and $\overline{t_i}$ is the current time (with respect to the sampling frequency).

Combining the RIRs. Each pressure-time function is passed through a band-pass filter whose cut-off frequencies are defined with respect to the central frequency of each octave. This way, only the spectrum corresponding to each band is extracted for each RIR. Finally, all the individual RIRs are combined in order to form the final RIR.

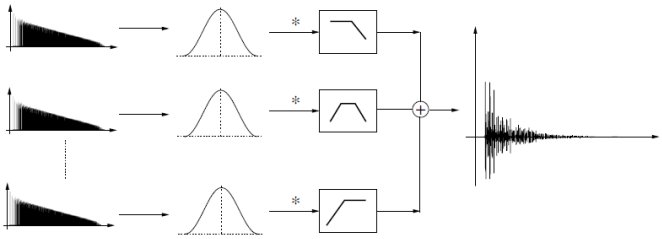

Fig. 6. The process of converting each echogram into pressure signals, filtering them, and obtaining the combined RIR (source: [37])

Filtering the RIR. The resulting RIR is similar in shape to a physically measured RIR. We performed spectral analysis of the two RIRs in Audacity [52] and concluded that the computer-generated one had a relatively constant intensity level throughout its audible spectrum. The physically measured RIR only exhibited this characteristic in a particular band, namely from 300Hz to 6000Hz, the intensity levels decaying exponentially outside this range. As such, a set of low-pass and band-pass filters were implemented in order to attenuate the frequencies outside the aforementioned band. This brought the audible properties of the computer-generated RIR closer to the physically measured RIR's characteristics.

Obtaining the reverberated sound. The above-mentioned RIR is typically used with HRTFs in order to compute the BRIR, but it can also be used to directly compute the reverberated sound through convolution with an anechoic audio sample. If such a step is undertaken, the resulting sound can be used to convey information regarding the size of the simulated environment, but it does not possess any information regarding sound directionality, as the output signal is still single-channel (mono).

Sound directionality using HRTFs. In order to compute sound directionality, the direction between source and receiver must be specified. This is done by computing the azimuth and elevation angles for the receiver relative to the source. Considering only one direction for each sound source would represent the simplest implementation, but a series of important problems would quickly arise. Firstly, as it can be seen in Fig. 7 (left), specifying the direction of the sound if the path between source and receiver is obstructed would lead to inaccurate results. Also, as Fig. 7 (right) illustrates, even when a direct path exists, in order to improve the auralization result, multiple acoustic paths, such as those caused by reflections (indirect acoustic paths), need to be taken into account.

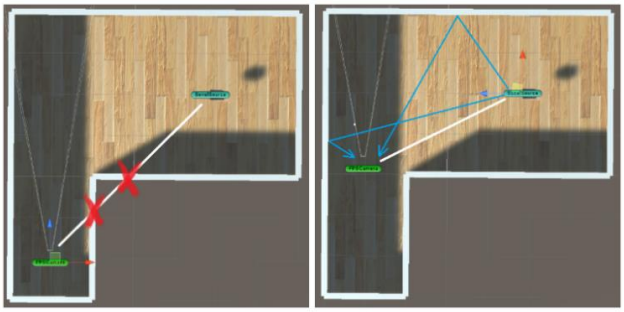

Fig. 7. Problems which arise when specifying only one sound direction per source. In the left image, the path between the receiver (green) and source (turquoise) is obstructed, and in the right image, reflected acoustic paths need to be taken into consideration

Solutions for the aforementioned problems have been identified in literature. In [53] and [54] the receiver is modeled as a sphere, where each face of the sphere represents an accumulation bin – thus producing a set of echograms for each considered direction. This way, multiple sound directions can be estimated by looking at the energies of the echograms in each bin. Paper [55] uses a cubemap to accumulate acoustical energy. Such solutions are not applicable in our pipeline because we calculate only one set of echograms for each iteration of the simulation (thus lacking any directionality characteristics), and also, the receiver's head is modeled as a point in space.

Nevertheless, a novel solution to resolve this issue has been implemented. In order to estimate sound arriving from multiple directions, the result of the Sonel Tracing algorithm has been reused. This result is represented by the computed sonel map, which in our case is the voxel map which stores the sonels. As the path of the sonels' emission depends on the scene's configuration, some regions will have voxels which contain a larger number of sonels – therefore, a higher acoustical energy. A heat map of this acoustical energy can be observed in Fig. 8, where the energy is color-coded. The assumption can be made that from these regions of higher acoustical energy more of the incident sound will also be reflected.

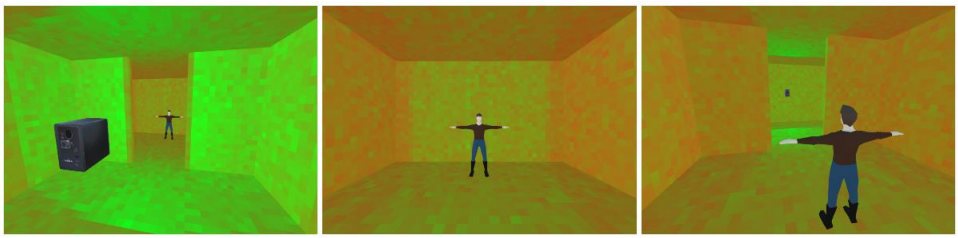

Fig. 8. Heat map of the acoustical energy stored in voxels, mapped from red (low energy) to green (high energy). View from behind the source (left), halfway between source and receiver (middle) and from behind the receiver (right). In the middle image, the acoustical shadow can be observed on the walls left and right to the receiver

In order to identify these regions of acoustical energy, a set of rays, which we call HRTF estimator rays, are traced through the scene. The rays' emission directions were computed using the Fibonacci lattice algorithm [56]. These intersect the scene's geometry, and using the number of sonels stored in each voxel, a subset of rays, which intersect the regions of highest energy, are selected. Afterwards, using the identified acoustical energy, as well as the distance from the receiver to the ray's intersection point, a normalized weight is computed for each direction, such that the sum of weights equals 1. Fig. 9 showcases such a scenario.

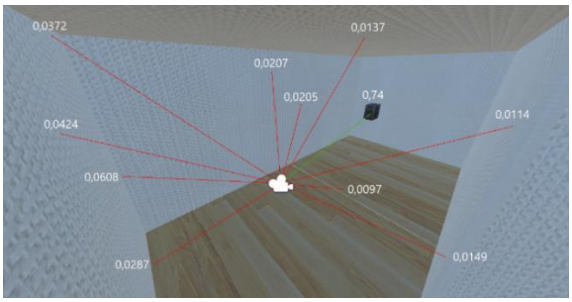

Fig. 9. The subset selected HRTF estimator rays: the weight for each ray has been overlayed

Finally, the spherical coordinates for each HRTF estimator ray are computed, and using the above-mentioned weights, a HRTF filter is selected for each direction and weighted accordingly. The final HRTF is constructed by summing up all individual weighted HRTFs. This final filter will contain a contribution from each of the computed directions.

$$HRTF_{final} = \sum_{i=1}^{N} HRTF_i \cdot w_i, \tag{5}$$

where N is number of HRTF estimator rays in the selected subset, $HRTF_i$ is the HRTF for estimator ray i, and w_i the weight for estimator ray i, for $0<w_i<1$.

5. Evaluation

In order to evaluate the functionalities proposed in our solution, a series of experiments were carried out. These aim to verify the methods which compute sound directionality. For each test, a number of audio samples were generated using the implemented auralization algorithms. These samples were emitted from different positions around the receiver and the participants were tasked with specifying the direction for each audio clip. For each experiment, absorbent materials were assigned to the surfaces of the scene in order to minimise the effects of indirect acoustic paths.

Estimating sound directionality from fixed emission angles. In this experiment participants were given 2 sets of audio samples. The first set contained 8 samples emitted at different azimuth angles, Fig. 10 (left) illustrates the virtual configuration. The second consisted of 4 samples emitted at different elevation angles, Fig. 10 (right) showcases the proposed configuration. In total, 16 participants took part in this experiment.

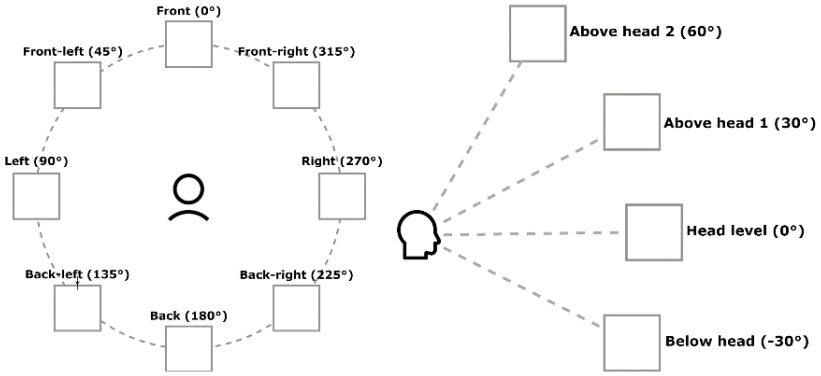

Fig. 10: Configurations used for azimuth (left) and elevation (right) angle estimation

Table 1

Results corresponding to the setup presented in Fig. 10 (left)

Azimuth angle	0°	45°	90°	135°	180°	225°	270°	315°
Localization accuracy [%]	87.5	31.25	68.75	25	75	31.25	12.5	31.25

Table 2

Results corresponding to the setup presented in Fig. 10 (right)

Elevation angle	-30°	0°	30°	60°
Localization accuracy [%]	18.75	43.75	12.5	43.75

The results presented in Table 1 and Table 2 illustrate the inability of sound localization from fixed angles. This occurs because the ambiguity of sound direction can be resolved by small movements of the head. In our experiment, sounds were emitted from fixed angles and thus direction ambiguity could not be solved. As such, a second experiment was elaborated, which evaluates sound directionality of a moving source.

Estimating sound directionality by panning the source. In this experiment, participants were given one set of audio samples. Each sample started playing at a fixed azimuth angle of x°, where x° can be any of the angles in Fig. 10 (left), and by the end of the sample, the sound source was rotated to an angle of x°+45°. In total, 7 participants took part in this experiment.

As Table 3 showcases, the results for this setup are more accurate in contrast with the results of the previous experiment. Participants were able to accurately identify the source's position when it was panned around the receiver.

Table 3

Results corresponding to the setup in which the audio source is panned around the receiver

Azimuth angle sweep	0° to 45°	45° to 90°	90° to 135°	135° to 180°	180° to 225°	225° to 270°	270° to 315°	315° to 360°
Localization accuracy [%]	85.71	100	100	85.71	85.71	100	100	85.71

The results are favourable enough to confirm that sound directionality was correctly implemented in the proposed solution.

6. Conclusions

In this paper we first presented the theoretical aspects relevant for acoustical simulators and thoroughly studied the methods used in literature. Using this knowledge, we decided on implementing a solution based on Sonel Mapping. We extended and improved upon its core functionalities by providing an intuitive way of prototyping virtual scenes, as well as integrating a method capable of performing auralization. In order to solve issues regarding the virtual scene's configuration, we developed a technique to quickly estimate sound directionality from direct and indirect acoustic paths. Finally, sound directionality was evaluated, which confirmed the validity of its implementation.

We have identified several constraints regarding our proposed solution. As a result of the voxelization and Sonel Tracing algorithms, the scene geometry (including sound sources) must remain static after running the acoustical simulation. For each adjustment brought upon the scene's configuration, these steps need to be executed again. Also, we have identified that the algorithms used for the acoustical simulation, alongside the signal processing methods employed for auralization, require a significant amount of computation time to complete. Regardless of the specified parameters, the simulation is computed in the order of seconds, but this prohibits us from running our solution close to real-time requirements. As such, these techniques should be reimplemented to allow for GPU acceleration using Compute Shaders. Furthermore, we have only considered the effects of absorption, specular and diffuse reflections. A diffraction mechanism should be implemented in order to improve the results of the simulation. The effect of diffraction can be significant especially at lower frequencies, in which case the wavelength is comparable in size with the virtual scene's geometry. It can also be noted that the building mechanism, which allows for the construction of the virtual environment, is well suited for further development. Currently, the provided implementation only represents a tool capable of quickly prototyping simple scenes, in which walls must be perpendicular to the ground and of the same height. Additional developments of this module might be useful in more complex scenes.

The solution was evaluated in terms of sound directionality. The second experiment provided us with valuable insight with regard to the validity of our implementation. In order to evaluate the solution as a whole, further research should be carried out in order to compare our implementation with other solutions presented in literature. The accuracy of the acoustical simulation as well as computation times can thus be evaluated. A suitable approach which can be analysed is Resonance Audio [57], as it provides similar functionalities and support for the Unity engine. These evaluations should provide us with insight regarding the performance of our solution.

As a whole, the proposed pipeline can be considered complete. It allows a user to start from a clean slate, build a desired environment using the provided mechanism, and perform the acoustical simulation. The results of this simulation can ultimately be used to allow the user to acoustically experience the virtual environment.

REFERENCES

[1] M. Vorländer, Auralization. Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality, Heidelberg: Springer-Verlag, 2008.

[2] B. Kapralos, M. Jenkin and E. Milios, "The Sonel Mapping Acoustical Modeling Method - Technical Report CSE-2006-10," York University, Toronto, 2006.

[3] M. Born and E. Wolf, Principles of Optics, Cambridge University Press, 1999.

[4] S. B. Shelley, "Diffuse boundary modelling in the digital waveguide mesh," Universuty of York, York, 2007.

[5] P. Charalampous and D. Michael-Grigoriou, "Sound propagation in 3D spaces using computer graphics techniques," in Proceedings of the 2014 International Conference on Virtual Systems and Multimedia, VSMM 2014, 2014.

[6] Z. He, A. Cheng, G. Zhang, Z. Zhong and G. Liu, "Dispersion error reduction for acoustic problems using the edge-based smoothed finite element method (es-fem)," International Journal for Numerical, vol. 88, no. 11, pp. 1322-1338, 2011.

[7] S. Kirkup, "The Boundary Element Method in Acoustics: A Survey," Applied Sciences, vol. 9, no. 8, p. 1642, 2019.

[8] L. Savioja, "Real-time 3d finite-difference time-domain simulation of low-and mid-frequency room acoustics," 13th Int. Conf on Digital Audio Effects, vol. 1, p. 75, 2010.

[9] N. Raghuvanshi, "Interactive physically-based sound simulation," University of North Carolina, 2010.

[10] L. Savioja and U. P. Svensson, "Overview of geometrical room acoustic modeling techniques," he Journal of the Acoustical Society of America, vol. 138, no. 2, pp. 708-730, 2015.

[11] R. Mehra, N. Raghuvanshi, L. Antani and D. Manocha, "A real-time sound propagation system for noise prediction in outdoor spaces," INTER-NOISE and NOISE-CON Congress and Conference Proceedings, vol. 2012, no. 4, pp. 7026-7035, 2012.

[12] R. Mehra, N. Raghuvanshi, L. Savioja, M. C. Lin and D. Manocha, "An efficient gpu-based time domain solver for the acoustic wave equation," Applied Acoustics, vol. 73, no. 2, pp. 83-94, 2012.

[13] R. Mehra, N. Raghuvanshi, L. Antani, A. Chandak and S. Curtis, "Wave-based sound propagation in large open scenes using an equivalent source formulation," ACM Transactions on Graphics (TOG), vol. 32, no. 2, p. 19, 2013.

[14] M. Zamith, E. Passos, D. Brandão and A. A. Montenegro, "Sound Wave Propagation Applied in

Games," in 2010 Brazilian Symposium on Games and Digital Entertainment, 2010.

[15] T. Reuben, "Wayverb - Context," [Online]. Available: https://reuk.github.io/wayverb/context.html.

[16] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," The Journal of the Acoustical Society of America, vol. 65, p. 943, 1979.

[17] F. Mechel, "Improved mirror source method in room acoustics," Journal of sound and vibration, vol. 256, no. 5, pp. 873-940, 2002.

[18] D. Schröder and T. Lentz, "Real-time processing of image sources using binary space partitioning," Journal of the Audio Engineering Society, vol. 54, no. 8, pp. 604-619, 2006.

[19] A. Farina, "Ramsete - a new pyramid tracer for medium and large scale acoustic problems," Proc. of Euro-Noise, vol. 95, 1995.

[20] M. Sikora, I. Mateljan and N. Bogunović, "Beam tracing with refraction," Archives of Acoustics, vol. 37, no. 3, pp. 301-316, 2012.

[21] M. Sikora and I. Mateljan, "A method for speeding up beam-tracing simulation using thread-level parallelization," Engineering with Computers, pp. 1-10, 2013.

[22] A. Krokstad, S. Strom and S. Sørsdal, "Calculating the acoustical room response by the use of a ray tracing technique," Journal of Sound and Vibration, vol. 8, no. 1, pp. 118-125, 1968.

[23] M. Okada, T. Onoye and W. Kobayashi, "A ray tracing simulation of sound diffraction based on the analytic secondary source model," Audio, Speech, and Language Processing, IEEE Transactions, vol. 20, no. 9, pp. 2448-2460, 2012.

[24] M. Dreher, G. Dutilleux and F. Junker, "Optimized 3d ray tracing algorithm for environmental acoustic studies," in Proceedings of the Acoustics 2012 Nantes Conference, Nantes, 2012.

[25] C. Schissler, R. Mehra and D. Manocha, "High-order diffraction and diffuse reflections for interactive sound propagation in large environments," ACM Transactions on Graphics, vol. 33, no. 4, pp. 1-12, 2014.

[26] M. Bertram, E. Deines, J. Mohring, J. Jegorovs and H. Hagen, "Phonon tracing for auralization and visualization of sound," VIS 05. IEEE Visualization, 2005., pp. 151-158, 2005.

[27] B. Kapralos, M. Jenkin and E. Milios, "Sonel Mapping: A Probabilistic Acoustical Modeling Method," Building Acoustics, 2008.

[28] C. Lauterbach, A. Chandak and D. Manocha, "nteractive sound rendering in complex and dynamic scenes using frustum tracing," Visualization and Computer Graphics, IEEE Transactions, vol. 13, no. 6, p. 1672–1679, 2007..

[29] A. Chandak, C. Lauterbach, M. Taylor, Z. Ren and D. Manocha, "Adfrustum: Adaptive frustum tracing for interactive sound propagation," Visualization and Computer Graphics, IEEE Transactions, vol. 14, no. 6, p. 1707–1722, 2008.

[30] M. Taylor, A. Chandak, Z. Ren, C. Lauterbach and D. Manocha, "Fast edge-diffraction for sound propagation in complex virtual environments," EAA auralization symposium, pp. 15-17, 2009.

[31] "VRWorks - Audio," NVIDIA Developer, 2019. [Online]. Available: https://developer.nvidia.com/vrworks/vrworks-audio. [Accessed June 2020].

[32] "Steam Audio," [Online]. Available: https://valvesoftware.github.io/steam-audio/.

[33] H. W. Jensen, Realistic Image Synthesis Using Photon Mapping, New York: A K Peters/CRC Press, 2001.

[34] "Reverberation Time," [Online]. Available: http://230nsc1.phy-astr.gsu.edu/hbase/Acoustic/revtim.html.

[35] "Reverberation Time RT60 Measurement," [Online]. Available: https://www.nti-audio.com/en/applications/room-building-acoustics/reverberation-time-rt60-measurement. [Accessed February 2021].

[36] L. Cao, Q. Fu, Y. Si, B. Ding and J. Yu, "Porous materials for sound absorption," Composites Communications, vol. 10, pp. 25-35, 2018.

[37] N. Tsingos and J.-D. Gascuel, "Acoustic simulation using hierarchical time-varying radiant exchanges,"

2005.

[38] N. J. P. Postma, "Serious auralizations (Auralisations sérieuses)," University of Paris-Saclay, Paris, 2017.

[39] H. Kutruff, "Auralization of impulse responses modelled on the basis of ray-tracing results," J. Audio Eng. Soc., vol. 41, no. 11, pp. 876-880, 1993.

[40] H. Kutruff, Room acoustics, 4th ed., Londra: Applied Science Publishers, 2000.

[41] B. Dalenbäck., P. Svensson and M. Kleiner, "Room acoustic prediction and auralization based on an extended image source model," J. Acoust. Soc. Am., vol. 92, no. 4, 1992.

[42] J. W. Strutt, "On our perception of sound direction," Philosophical Magazine, 1907.

[43] D. W. Batteau, "The Role of the Pinna in Human Localization," Proceedings of the Royal Society of London. Series B, Biological Sciences, vol. 168, no. 1011, pp. 158-180, 1967.

[44] "HRTF-Database," [Online]. Available: https://www.oeaw.ac.at/isf/das-institut/software/hrtf-database. [Accessed May 2021].

[45] M. Förster, "Bachelor's Thesis - Auralization in Room Acoustics," Graz University of Technology, Graz, 2008.

[46] S. Pelzer, L. Aspöck, D. Schröder and M. Vorländer, "Integrating Real-Time Room Acoustics Simulation into a CAD Modeling Software to Enhance the Architectural Design Process," Buildings, vol. 4, pp. 113-138, 2014.

[47] T. Lentz, D. Schröder, M. Vorlaender and I. Assenmacher, "Virtual Reality System with Integrated Sound Field Simulation and Reproduction," EURASIP Journal on Advances in Signal Processing, vol. 2007, no. 1, 2007.

[48] "Chordless Cycle," [Online]. Available: https://mathworld.wolfram.com/ChordlessCycle.html.

[49] "Sengspiel Audio - Sound power level SWL and sound pressure level SPL," [Online]. Available: http://www.sengpielaudio.com/calculator-soundpower.htm. [Accessed June 2020].

[50] "Sengspiel Audio - Formula for calculating the damping of air," [Online]. Available: http://www.sengpielaudio.com/AirdampingFormula.htm.

[51] "Sound Speed in Gases," [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/souspe3.html. [Accessed June 2020].

[52] "Audacity," [Online]. Available: https://www.audacityteam.org/. [Accessed June 2021].

[53] S. Schimmel, M. Muller and N. Dillier, "A fast and accurate "shoebox" room acoustics simulator," IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, pp. 241-244, 2009.

[54] D. Schröder, Physically Based Real-Time Auralization of Interactive Virtual Environments, Ph. D., 2011.

[55] N. Röber, U. Kaminski and M. Masuch, "Ray acoustics using computer graphics technology," in Proceedings of the 10th International Conference on Digital Audio Effects, 2007.

[56] R. P. Stanley, "The Fibonacci Lattice," University of California, Berkeley, 1975.

[57] "Resonance Audio," [Online]. Available: https://resonance-audio.github.io/resonance-audio/. [Accessed July 2021].