LiU-ITN-TEK-A--22/006-SE

Immersive Audio - Simulated Acoustics for Interactive Experiences

Linus Arvidsson

2022-05-10

Department of Science and Technology
Linköping University
SE-601 74 Norrköping, Sweden

Institutionen för teknik och naturvetenskap
Linköpings universitet
601 74 Norrköping

LiU-ITN-TEK-A--22/006-SE

Immersive Audio - Simulated Acoustics for Interactive Experiences

The thesis work carried out in Medieteknik
at Tekniska högskolan at
Linköpings universitet

Linus Arvidsson

Norrköping 2022-05-10

Department of Science and Technology
Linköping University
SE-601 74 Norrköping, Sweden

Institutionen för teknik och naturvetenskap
Linköpings universitet
601 74 Norrköping

Abstract

A key aspect of immersive audio is realistic acoustics. To get plausible acoustics for an environment the impulse response can be generated using acoustic simulations. In theory, the impulse response changes depending on the location of the sound source and listener in the scene. The impulse response should therefore ideally be updated in real-time for interactive applications which require fast acoustic simulation methods like ray tracing.

In this thesis the listening experience of sound generated with an interactive sound propagation engine was explored and compared to spatial sound produced with a static impulse response. The aim was to evaluate the sound experience for applications outside of virtual reality, with computational cost in consideration. This was done by conducting a user study where the participants got to interact and compare the two sound methods in different environments. The study was performed using a custom developed application integrated with a pre-existing sound propagation engine.

The results from the user study showed no obvious perceptive difference between the two sound rendering methods that could justify the extra computations. Overall there was even a slight preference for the stereo method that used a static impulse response. However, there were qualities to both sound rendering methods that were preferred depending on the environment.

Another thing that was investigated in the work of this thesis was how the varying accuracy of localization of sound in different directions can be used in acoustic ray tracing algorithms. An alternative sampling method was developed that uses a biased distribution based on spatial resolution of human hearing instead of traditional uniform sampling. The computation time of the random sampling phase increases, but could potentially reduce the number of ray samples needed. Testing of this method was limited and could neither show any benefits or drawbacks with biased sampling.

Contents

List of Figures

1 Introduction

A commonplace goal of interactive experiences is to achieve immersion, whereby the user feels present in the presented world. Immersion can be achieved through different means which engage different parts of the brain. Techniques include the telling of engaging narratives or in the case of interactive experiences mechanics that generate sensory-motoric feedback loops. The more direct way is to try to achieve spatial immersion by introducing realistic stimuli that mediates a convincing world adhering to the physical rules that the mind is trained to detect. Spatial information is primarily perceived using sight and hearing. While advances in the technology have led to many significant graphical improvements over the years, there has been less emphasis on more realistic audio.

Auralization[9] is the procedure of rendering audible samples that simulates the listening experience at a position in a modeled space. Doing real-time auralization for interactive media can be essential for spatial immersion and interest in realistic spatial audio has grown as virtual reality hardware has improved. Auralizations of a digital space require simulation of how sound propagates through the environment which requires heavy computations which are challenging to perform at interactive rates. While the benefits of interactive sound propagation have been explored for virtual reality applications, there has not been as much research into the benefits for traditional flat screen experiences.

There are two prevailing approaches to simulating sound propagation for acoustics. One approach is to solve the wave equation numerically which gives accurate results but also requires very heavy computations. The other approach is to use what is called geometrical acoustics, or GA for short. With GA simulation sound waves are assumed to propagate as rays. This assumption can be considered physically correct for high frequencies where the wavelength is much smaller than the object but falters at lower frequencies where the wave properties become more important. Despite the accuracy concerns with GA, its simplicity makes it perfect for efficient acoustic simulations which is especially important for real-time applications.

1.1 Aim

The aim of the work in this thesis is to explore the potential benefits of geometric acoustic (GA) propagated sound simulation compared to a more conventional convolution reverb

method for interactive experiences outside of virtual reality. A pre-existing sound propagation engine will be integrated into a game environment where a user study will be conducted to evaluate the listening experience. Some further optimizations of the simulation will also be tested utilizing the psychoacoustic property of spatial resolution. The sound propagation engine that will be used in the tests is Gsound[20] developed by Carl Schissler at University of North Carolina at Chapel Hill.

1.2 Research questions

- How is interactive sound propagation experienced compared to more traditional stereo methods for games?
- Is there a major perceptive difference with interactive sound propagation compared to a static reverb for applications outside of VR that justifies the computational cost?
- Can the variation of spatial resolution for different angles be utilized in acoustic simulations to minimize computational load?

1.3 Delimitations

This work done in this thesis is limited to a geometrical acoustic implementation for sound propagation and does not cover any wave-based solutions. Although wave-based algorithms have some advantages, they require complex computations that need to be pre-calculated for a real-time application. Geometric solutions are more straightforward to implement and can easily handle dynamic scenes with moving sound sources. There are some commercial sound propagation plugins like Steam Audio and Oculus Audio for game engines such as Unreal Engine, but they were not used due to their closed nature. Another method that was not covered in this thesis is the use of head-related transfer functions (HRTFs). HRTFs would make it possible to create more realistic spatialized audio but would also add a lot of complexity to both the sound implementation and user study.

2 Theory

There are many physical aspects to take into account when creating realistic sound for an interactive 3D-environment. Two easy effects to use when spatializing audio are panning and fading. Panning between the left and right channel in a stereo setup makes it possible to tell what direction a sound comes from while fading the volume up or down gives the listener ques of the distance to the sound source. An equally important thing to consider is acoustics which can be approximated by adding a simple reverberation effect. However, a common problem is that the reverberation effect does not match the visuals and does not meet the listener's expectation of how things should sound. For a more immersive experience the sound should be more realistic and consider how sound waves interact with the environment. A good approach to this is by using an impulse response.

2.1 The room impulse response

Adding acoustic information to an audio signal is commonly done utilizing an impulse response[16]. The room impulse response is the transfer function between the sound source and microphone and contains the acoustic characteristics for a specific point in the location. As the name implies the input signal is a unit impulse which in theory would be an infinitesimally short, extremely loud sound. In practice the impulse can be approximated with something like the bang of a gun. A perfect impulse signal contains all frequencies with equal amplitude which is hard to replicate with a bang. This property does however enable the use of frequency sweeps where the frequencies detectable by the human ear are played separately in a sinusoidal sweep which usually give more accurate results. The recorded response is then processed using deconvolution to create the impulse response.

The room impulse response is the sound energy response from the impulse sound that has reached the listening position in a room. Due to the sound attenuating through the air and the interaction with the environment, the sound waves will arrive at different times with different amounts of energy. This is what makes the room impulse response an acoustic description of a specific listener position in a location. The acoustic sound signature of the room can then be added to any sound signal by convolving it with the impulse response in what is called convolution reverb. Although convolution can be performed in the time domain, it is far more efficient to convolve in the frequency domain considering the temporal length of

audio clips. This is done by first transforming the input using a fast Fourier transform (FFT) algorithm.

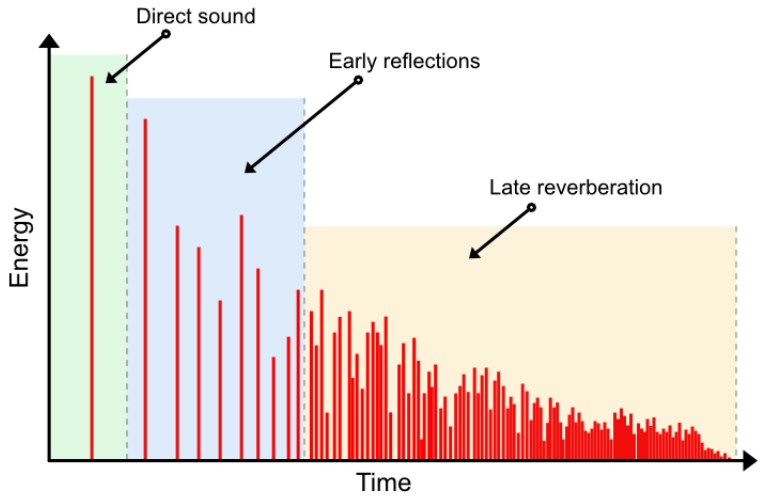


Figure 2.1: A simplified graph of a time-energy response from an impulse sound showing the layout of the three different categories of energy spikes.

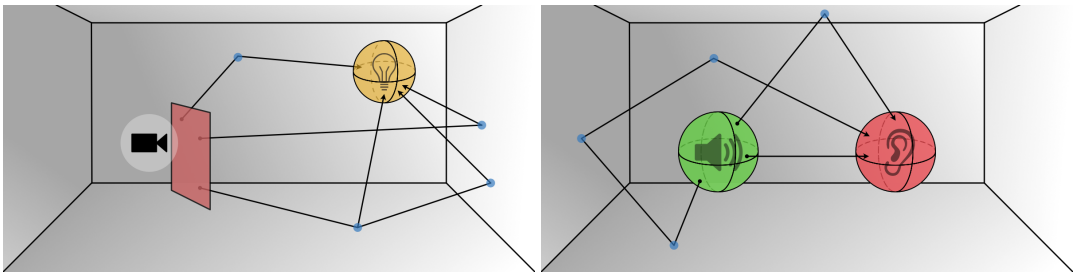
Each peak in the plot of an impulse response represents a reflection path of the sound wave. The room impulse response is usually divided into three distinct parts with different characteristics[23], see Figure 2.1. The first spike is the direct sound which is only delayed by the distance from the sound source. What follows are the early reflections from the environment that arrive close enough in time so that they can not be perceived separately from the direct sound because of the Haas effect. It is these two categories of sound that present the most information about the source such as its distance and loudness. The last part is the late reverberation consisting of higher-order reflections from the environment. The high amount of reflection creates a diffuse sound field where individual echoes cannot be heard. Reverberation is the part of sound that is most associated with the acoustic properties of a room and gives hints to things like the scale and shape of the room.

2.2 Geometrical acoustic simulation

With auralization of a digital scene the impulse response has to be generated through an acoustic simulation. Efficient acoustic simulations usually build on the assumption that sound waves propagate as rays to enable faster computations in what is called geometrical acoustics (GA)[18]. This assumption can be considered physically correct for high frequencies where the wavelength is much smaller than the object but falters at lower frequencies where the wave properties become more important. These wave properties can however be approximated using various geometrical methods.

Ray tracing sound is in many ways similar to ray traced global illumination in graphics rendering, see Figure 2.2. The main difference being that the speed of light is fast enough to be considered instantaneous for rendering purposes while the relatively slow speed of sound is what gives rise to most acoustic effects. Geometrical acoustic ray tracing algorithms are used to find valid sound propagation paths between the source and the listener. Usually the sound propagation simulation is divided into the three distinct categories of direct sound, early reflections and late reverberation, each using different approaches to find the propagation paths. Frequency dependent effects related to reflections and absorption are usually added by performing the simulation in several discrete frequency bands. The energy can then be

scaled for different frequencies using absorption coefficients. With the propagation paths found the resulting sound can be rendered either by rendering each valid path individually and combining them or by synthesizing an impulse response and using convolution.



(a) Global illumination (b) Sound propagation

Figure 2.2: An illustration of ray tracing for global illumination and sound propagation. Light is most commonly traced backwards from the camera in a frustum with the end result being a flat image. Sound is usually cast in random directions from a source and is then traced until it hits a spherical detector.

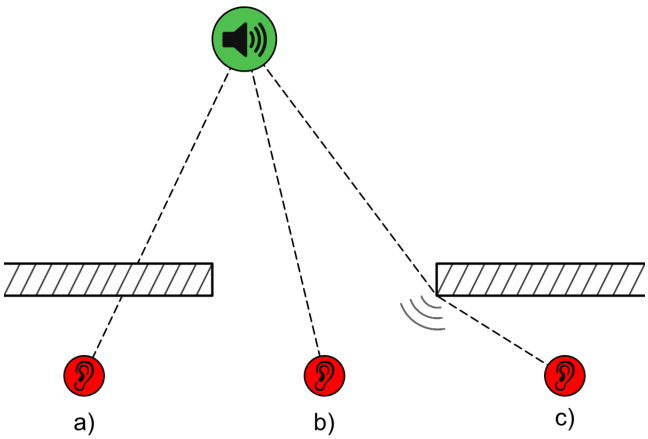


Figure 2.3: Three different non-reflecting propagation paths. a) The sound is transmitted through the wall reaching the listener. b) The sound reaches the listener in a direct path. c) The sound is diffracted on the edge

2.2.1 Direct sound

There are several categories of propagation paths sound can take to reach a listener without reflecting on any surface, see Figure 2.3. The simplest propagation path in acoustic simulations is direct sound where the sound wave can travel from the sound source to the listener without any interaction with scene geometry. To check if the path between the source and the listener is unobstructed a visibility ray can be cast. If there are no triangle intersections the path is clear and can form a valid direct path. In an unobstructed path the only energy that is lost is through the sound attenuation for the propagating medium. If the source and listener are defined as points one visibility ray is cast for every source in the scene. For larger spherical sources or detectors multiple visibility rays can be cast from different points of the sphere to calculate a visibility factor that is used to scale the direct sound spike[27]. Rays are sampled using random directions defined within a cone defined between the view point and

the outline of the detection sphere. The visibility factor is calculated as the number of valid visibility paths divided by the total number of visibility rays cast for the source.

2.2.2 Diffraction

Another physical effect to consider for propagation paths is diffraction whereby sound waves bends around the edges of an obstacle. An obvious case of this would be sound radiating from the door opening of a room. For ray implementations, diffraction can be approximated as a piecewise-linear propagation path using the unified theory of diffraction (UTD)[10]. UTD has been adapted to real-time simulations [26], but was for long limited to static scenes. Later implementations of UTD utilizes a preprocessed edge visibility graph to achieve higher order diffraction paths in dynamic scenes[22]. In the preprocessing phase the edges of a mesh are classified as either a diffracting or non-diffracting edge based on the angle to the neighboring triangles. The diffracting edges are then compared to other diffraction edges to see if they are visible to each other and can form a diffraction path which will be added to the visibility graph. When a ray then intersects with a triangle its edges are checked if they are classified as diffraction edges. The visibility graph can then be used to form diffraction paths for that edge.

An alternative and more accurate but also more expensive approach to edge diffraction is the Biot-Tolstoy-Medwin (BTM) method[25], which uses line integration over the diffraction edges, based on the work of Biot and Tolstoy[2] and Medwin[13]. A completely different approach is to base the calculation of diffraction on Heisenberg's uncertainty principle[24]. This method can be combined with ray tracing and works differently with the assumption that the diffraction effect gets stronger the closer the ray is to an edge.

2.2.3 Specular reflections

Some parts of propagating sound waves that interact with the environment will reflect in a specular fashion according to the law of reflection which states that the incident angle equals the outgoing angle relative to the surface normal, see Figure 2.4. Specular reflections are especially prominent in the early reflections which play a big part in the perceptive localization of objects. Figure 2.5 shows specular sound paths that make it possible to hear a sound source that would otherwise be blocked.

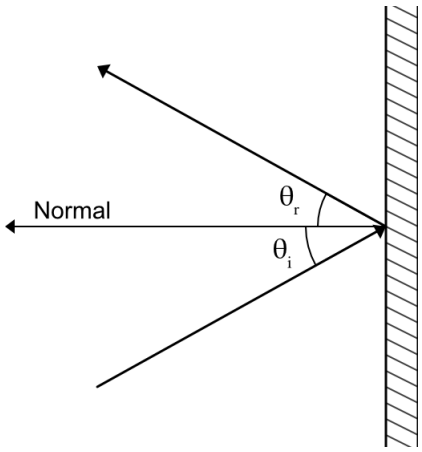


Figure 2.4: An ideally specular reflection. The incident ray comes in at an angle of θ_i relative to the normal and the outgoing ray an angle of θ_r where $\theta_i = \theta_r$.

A commonly used geometric method for specular reflections is the image source method[1]. The method works by mirroring the sound sources against all surfaces in the

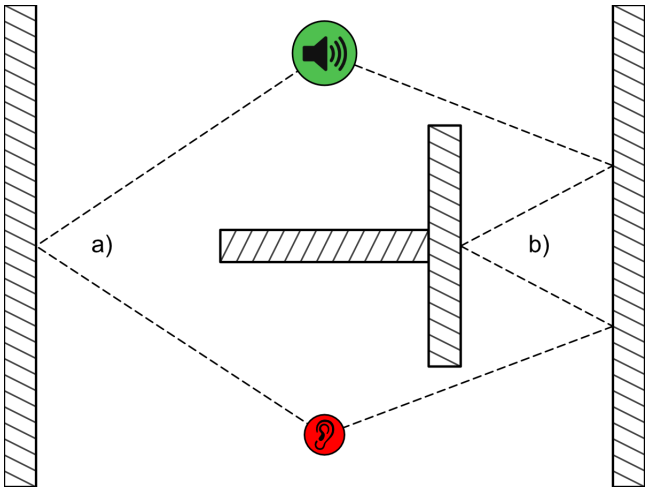


Figure 2.5: Example of the importance of specular reflections in a scene with two propagation paths that are only possible with reflections. Path a) illustrates a first order specular reflection and path b) Illustrates third order specular reflections with three bounces.

scene creating a set of image sources, see Figure 2.6. Higher order reflections can be considered by recursively mirroring the created image sources against the scene geometry. All potential reflection paths are then gathered by validating the paths between the listener and each image source in the scene where the path must intersect with each reflection surface for the image source but no other surfaces. The image source algorithm is very simple and can deterministically predict all ideally specular reflections but it can also be slow since it grows exponentially with reflection depth which gets especially bad for scenes with dense geometry.

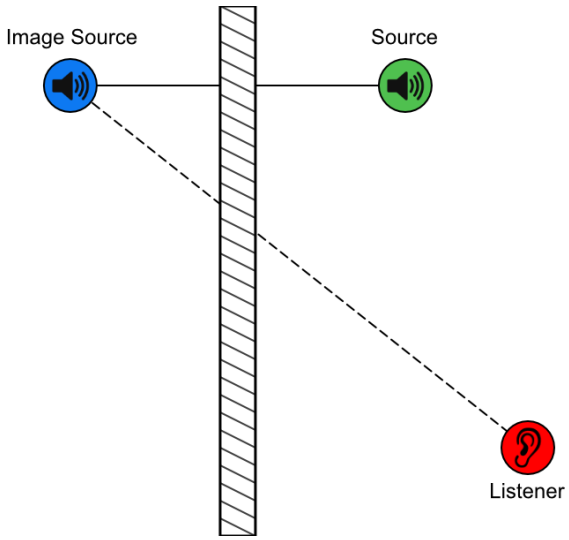


Figure 2.6: The image source method. The sound source is mirrored against all walls in a room to generate image sources. Validation of the image source is then done by testing the path to the listener position. The path needs to intersect the same wall that was used to mirror the image, and only that wall, to validate the image.

A big problem with the traditional image source method is that many of the generated image sources will not be valid in the end which leads to unnecessary computations. A way to mitigate this is by using a hybrid ray-tracing and image source method[27]. This is done

by tracing rays through the scene, noting the surfaces that they hit on the way, until they hit a detector. Knowing the order of surfaces hit by the rays it is possible to form the exact location of the image sources. Another optimization of the image source method for use in auralizations is through the use of beam tracing[7]. The beam tracing optimization uses culling with beams to remove invalid images earlier to limit the exponential growth of image sources in higher order reflections.

Another approach to specular reflections for auralizations is through ray tracing[11]. Using Monte carlo ray tracing a great number of rays are cast in random directions from the source position. To simulate specular reflections a new ray is cast from the intersection point of a surface in the direction calculated with law of reflection using the surface normal. When a ray hits a detector volume, representing the receiver or listener, it is registered as a valid path and used to form the impulse response. Unlike the image source method the Monte Carlo ray tracing approach will not deterministically consider every possible reflection path and requires a large amount of ray samples to get an accurate result. It can also be considered wasteful to trace paths that will never hit a detector and therefore does not have a contribution to the impulse response at the listening position. A way to minimize the number of unused traced sound paths is to instead do backward ray tracing which is based on observation that perceptually important propagation paths tend to come from the vicinity of the listener[20]. Combined with an image source method, listener images can be formed to from the intersected triangles to calculate exact specular paths to the sound source.

2.2.4 Diffuse reflections

Sound reflections are usually not perfectly specular since things like roughness in the reflecting material will scatter the sound in all directions, see Figure 2.7. Diffuse reflections play an especially important role for late reverberation which is dominated by scattered sound and high order reflections.[12] Deterministic algorithms using image source methods are not able to model diffuse reflections and can therefore not be used to create the late reverberation of the room impulse response. This is however something stochastic ray tracing methods can model. Ideal diffuse reflections where sound is scattered equally in all directions can be modeled by casting rays in random directions from the intersection point but this is not how sound interacts with most materials. Sound is usually reflected some part in a specular fashion and some part diffuse fashion scattered within a non-uniform distribution.

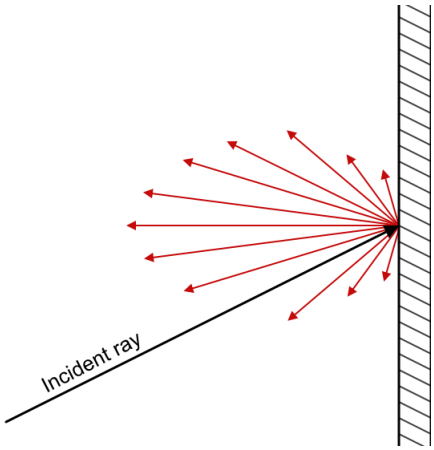


Figure 2.7: Example of a diffuse reflection. The material scatters the sound in all directions.

A simple way to implement scattered reflections would be to generate multiple reflecting rays from the intersection point including a specular reflection. This solution is however not feasible since the number of rays would grow exponentially with every reflection. A more

efficient way is to only reflect one ray using vector based scattering[6] where the reflected ray is a linear combination between the specular direction and a random lambertian distributed scattered direction. The amount of scattering in a material can then be specified using a scattering coefficient s which is used to scale the specular and scattered vectors, see Figure 2.8. The amount of scattering in a material is dependent on the frequency, with higher frequencies being prone to more scattering. The vector based scattering approach to this is to choose the scattering coefficient of a material for a mid-frequency.

A more complex way of capturing scattering behavior is with the use of a bidirectional reflectance distribution function (BRDF) for the reflecting material when calculating the new ray direction. BRDFs are commonly used in computer graphics to simulate how light reflects at a surface given an incoming and outgoing direction but can similarly be used for sound purposes with pre-computed acoustic BRDFs for ray tracing applications.[15]

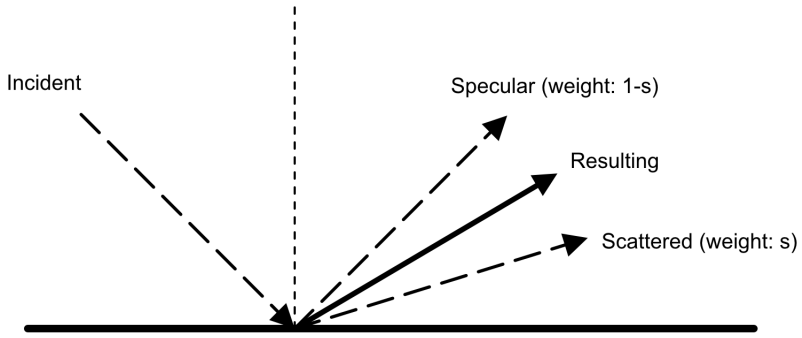


Figure 2.8: Vector based scattering. The resulting reflected vector is a linear combination of a specularly reflected and a diffusely reflected vector scaled with the weights $1 - s$ and s respectively, where s is the scattering coefficient.

One way to increase the number of propagation paths for late reverberation is by using diffuse rain sampling [23], see Figure 2.9. At every intersection point an additional ray is cast towards the detector, forming a new path if visible. The contributing sound energy of the path depends on the material properties at the intersection point and the hit probability of the detector. Solving the probability density function integral analytically for arbitrary angles is too costly and can instead be approximated by calculating the distribution for fixed angles between the surface normal and the connecting vector pointing towards the detector. Diffuse rain sampling can be performed using forward ray tracing where the detector is the listener or backward ray tracing [21] where the detector is the sound source.

A more robust solution to statistical methods like diffuse rain is to use more than one type of estimator with multiple importance sampling. By using bidirectional path tracing combined with multiple importance sampling to generate paths between the source and listener more accurate results can be achieved compared to the diffuse rain methods.[4] The algorithm works by first tracing rays from both the source and listener up to a set number of bounces. After that comes a connecting step which connects every intersection point of the forward tracing to the points of the backward tracing as well as the detectors of both paths. Similar to diffuse rain, the hit probability of each path is calculated to get its contribution of sound energy in the scene.

A completely different approach to geometric acoustics is the radiosity method which is often used in computer graphics for global illumination applications but can similarly be used for acoustic purposes with sound energy and the addition of temporal information. With radiosity the geometry in the scene is divided into larger surface patches that can store incoming sound energy. Sound energy is initially propagated from a sound source to the scene geometry where the amount of energy received at a surface patch is calculated using a

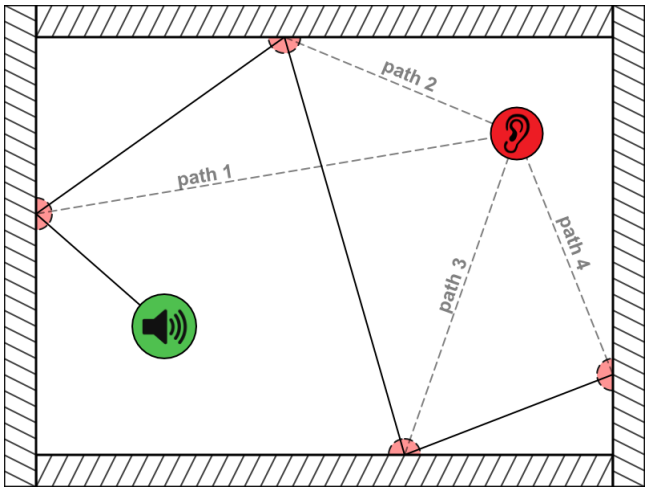


Figure 2.9: The diffuse rain algorithm. More propagation paths are generated by adding a path to the detector at every intersection point. The energy contribution of the path is calculated using the material properties of the surface of the intersection point as well as the position and size of the detector.

view factor. After that the energy received at each surface patch is iteratively distributed in the same way between all surface patches, simulating higher order reflections. The iterative energy distribution stops when the energy at every patch is lower than a set threshold which happens due to absorption. Energy is then gathered from the surfaces viewed from the listening position to generate an impulse response. Radiosity assumes all reflections to be ideally diffuse which is the opposite of an image source method which assumes all reflections to be specular.

2.3 Interactive sound propagation

Auralizations are usually done by first performing an acoustic simulation to generate the room impulse response which can then be used to render audio for the modeled space using convolution. A limitation of this method is that the generated impulse response is in theory only valid for a single point in the room with a fixed relationship between the listener and the sound sources. If however the acoustic simulation is updated in real-time, at the same time as the audio rendering, accurate acoustics with dynamic scenes can be achieved. With interactive sound propagation the room impulse response should be regenerated when the state of the listener has changed, making the auralization valid for every point in the scene. This means that the listener can move around in the scene and get the correct acoustic response at every position. It can also be extended to work for moving sound sources.

The simulations used for interactive sound propagation builds upon the knowledge of the auralization methods in the architectural acoustic field. Acoustic simulations can be done either by solving the wave equation numerically or by using geometrical acoustic methods that build on the assumption that sound propagates as rays. Geometric acoustic methods are typically preferred for interactive applications due to its speed. In contrast, solving the wave equation is very complex, requiring long computation times not suitable for interactive sound propagation. To achieve interactive auralizartions with wave-based simulation much of the calculations has to either be precomputed[5], or heavily simplified by for example doing real-time wave simulation in 2D[17] which removes the complexity of 3D-simulation. When it comes to real-time sound simulation in dynamic scenes geometrical acoustic methods offer more flexible and feasible solutions for current hardware.

2.4 Psychoacoustics

An area relevant to sound propagation systems is psychoacoustics. Psychoacoustics is the scientific study of how physical limitations of the ear and the effects from the brain's processing affects the human perception of sound. Digital sound applications are often optimized using psychoacoustical effects and metrics, with the motivation that it is unnecessary to compute sound effects undetectable for a human listener. A common example of this is the MP3-codec which utilizes psychoacoustic modeling with effects like auditory masking to compress audio. Psychoacoustic optimizations can also be applied to acoustic simulations which become especially important for interactive applications. The aim of most optimizations of geometrical acoustic methods is to minimize the number of rays that needs to be traced by either eliminating propagation paths that will not be noticable early in the tracing step, or reusing old propagation paths that are still perceptually valid.

The localization of sound sources is one area where psychoacoustics become relevant[3]. Due to the form of the ears and their location on the head the accuracy of localization of sounds is different for different incident angles. Since the ears are parallel in the horizontal plane the spatial resolution for localization of sources is significantly higher in that plane compared to the vertical plane. In the horizontal plane the spatial resolution is best right in front of the listener with a localization accuracy of about 1°. For sources behind the spatial resolution is slightly lower at circa 5° degree accuracy. The localization accuracy in the horizontal plane is the worst directly to the right and left of the listener where the accuracy is about 10°. The spatial resolution of the vertical plane is significantly lower with an accuracy of about 20° straight above and below the listener.

2.5 Gsound

Gsound is a sound propagation engine for interactive applications developed by Carl Schissler first presented in 2011.[20] The engine has since been continuously developed, adding more features and optimizations. The main propagation method traces specular reflections from the listener using a combination of ray-tracing and the image source methods. Diffuse reflections are traced from the sources using radiosity-like subdivisions in the sampling to reduce noise.[22] The algorithm also utilizes spatial and temporal coherence to cache diffuse paths to be able to cast fewer rays. For scenes with multiple sources, the engine has a backward ray-tracing implementation of diffuse reflections to avoid linear scaling of computation time with the number of sources.[21] Another optimization for multiple sources is the use of source clustering which groups distant sources that are perceptually hard to distinguish. Diffraction is implemented using the uniform theory of diffraction optimized with an edge visibility graph and simplified geometry.[22] The impulse response generation is also optimized using temporal coherence with an impulse response cache which uses psychoacoustic metrics to adapt length of the impulse response.[19]

3 Method

To investigate how simulated acoustics affect the experience in interactive applications a user study was conducted. When interacting with acoustic environments, graphics are essential as a visual reference. For this an application was developed as a platform for the user study using OpenGL graphics. The focus was on sound rather than graphics, which were kept minimalistic to not be distracting. The application uses basic phong lighting with a forward rendering pipeline which limits the geometric complexity and number of light sources but was enough to produce the 3D-environments for the user study. Acoustic simulation was done through integration with the sound propagation engine Gsound developed by Carl Schissler.[20] The main purpose of the application is to easily be able to compare the simulated audio, using the integrated sound propagation engine, to a traditional spatial rendering solution using a pre-applied reverb.

3.1 Audio setup

The audio rendering in the application had to be able to render audio both from Gsound and the stereo implementation of spatial audio. To enable a good comparison in the user test it also had to be possible to change the rendering method at the press of a button. The API used to handle the output to the sound driver of the computer was libsoundio[8] which is a cross-platform solution that works consistently with a variety of different audio backends. A write callback function for sound output had to be provided by the application to fill the output buffer with audio samples on command. Mixing of the different audio inputs is handled by an audio renderer class in the application which contains the common sound buffer read by the soundio callback function, see Figure 3.2. The common sound buffer uses a circular buffer data structure, visualized in Figure 3.1, optimal for continuous read and write operations. The structure contains pointers to the read and write positions which are continuously chasing each other in real-time rendering. When a pointer reaches the end of the buffer it loops back around to the start, making it possible to perpetually perform read and write operations.

An important issue to take into consideration is the potential of the buffer being underrun or overrun. The buffer is underrun when the application does not provide it with enough audio information in the rendering time frame, which creates an audible gap in the output as a result. In a circular buffer this happens when the read position pointer is incremented past the write position pointer. When the application provides more information than can be

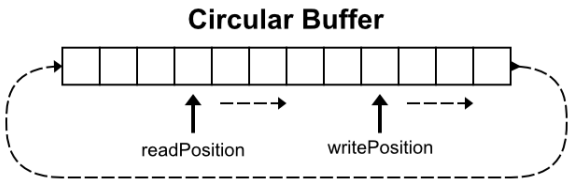


Figure 3.1: The circular buffer data structure. Audio data is written and read at the position of the corresponding pointer and loops around in a circular fashion writing over the old read data.

rendered in real-time the buffer becomes overrun causing a growing delay between the audio and what happens on screen.

To avoid an underrun or overrun buffer the application should ensure stable write updates with a consistent delta time. The audio renderer class gets updated from the main rendering loop of the application which uses a stable update time for a locked frame rate giving the audio updates a fixed delta time. Additionally Gsound renders audio on a seperate processing thread from the sound propagation and will render audio using the last propagated data if the latest propagation update cycle has not been finished. With a stable update time the choppy audio produced from when the read pointer catches up to the write pointer can be circumvented by adding a delay between the pointers that is slightly larger than the largest update time. In the long run small errors can potentially build up making the buffer unstable with this solution but a delay works well for the scope of the user tests.

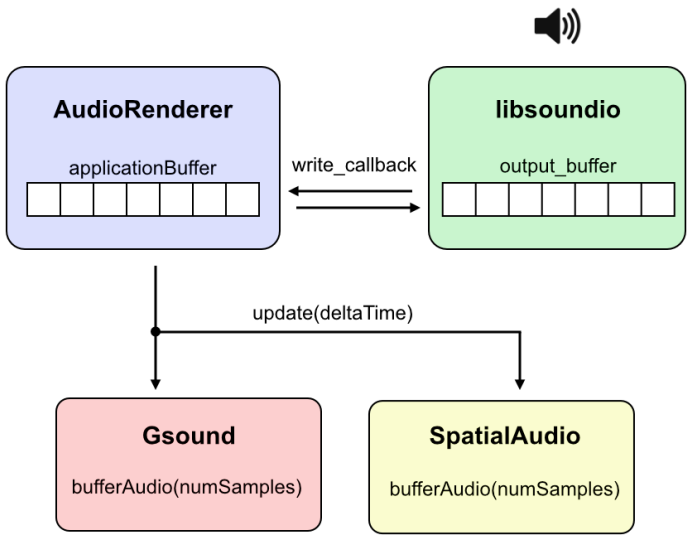


Figure 3.2: The flow of audio data in the application. The audio renderer class contains a common application buffer that gets updated with data from one of two spatial rendering solutions. This buffer provides the real time data for the write callback function connected to the libsoundio API that renders the audio to the sound driver of the computer.

Audio is loaded into the application and stored in the array of an audio object. Playback of this audio can then be done through an audio instance, which is a class that handles the reading of the audio samples and controls the volume. Loading all audio at the start of the application is not an ideal use of memory compared to progressively streaming it from the

secondary memory of the computer. However, for the purpose of this thesis preloading the audio works and is well within the memory budget of the application.

3.2 Spatial sound implementation

A simplified spatial sound system was developed with the purpose of approximating more traditional game rendering methods for comparison during the user study. This system is built up with a class containing sound source objects that stores a position with coordinates in world-space and audio instances for the playback. To spatialize audio from the sources there are several effects that should be present for a realistic representation. Acoustics are added using convolution reverb. Rather than doing real-time convolution the reverb is applied beforehand to the audio track. Each source has audio instances of both an audio track without reverb and a track with reverb, which makes it easy to dynamically mix the acoustic effects with a dry and wet parameter. For a fair comparison between the interactive sound propagation and this solution an impulse response is generated for each scene using the same simulation software. The positioning of the source and listener is crucial to get an acceptable approximation of the entire room. When generating the impulse responses for the scenes the source was placed at a central position with equal distance to the walls and the listener at a distance where the reverberation is dominant compared to the direct sound.

The distance to the source is mediated through an attenuation effect that simulates how the sound volume drops as it gets further away. In practice there are different types of drop-off curves that are typically used for this purpose depending on the use case. To get a natural sounding attenuation for this implementation, a logarithmic function was used. This makes it so that sounds are powerful close up followed by a quick drop off in volume with a slower drop off at greater distances. The approximated attenuation function used is defined as:

$$\text{attenuation} = 1 - c \cdot \log_{10}(\text{ distance}) \tag{3.1}$$

where c is a coefficient that can be used to change how fast the sound drops off. This coefficient is used to scale the dry and wet signal of the reverb as shown in Figure 3.3. The coefficients were selected by comparing the attenuation to that of the simulated propagation with Gsound. For distances less than one meter, the dry signal is kept at an attenuation value of one while the wet signal is faded out to with a closer distance get a cleaner sound close up to the source.

A sense of directionality is gained from panning the sound between the left and right channel. The panning scale factors for the channels are calculated using the dot product between a vector pointing to the left of the listener and a vector pointing towards the source from the listener as shown in Equation 3.2:

$$\text{left} = 0.5 + \frac{\vec{v}_{left} \cdot \vec{v}_{source}}{2} \tag{3.2a}$$

$$\text{right} = 0.5 - \frac{\vec{v}_{left} \cdot \vec{v}_{source}}{2} \tag{3.2b}$$

Sound reaches both ears even if the source is in a position straight to the left or right of the listener since sound waves go through and around the head. This would make panning for direct sound with volume values between 0 and 1 sound unnatural with silence in the ear opposite to the source location. To deal with this, only 90% of the volume of direct sound is affected by panning. The same problem arises with the reverberation signal. Since reverberation is the result from the sound waves bouncing around the room the directionality is much more diffuse than the direct sound. Because of this the panning is approximated to only affect 40% of the volume for the reverb signal in this implementation.

The final effect considered in the custom spatial sound system is sound obstruction. When a big object like a wall is blocking the sound the direct path should not be audible. This

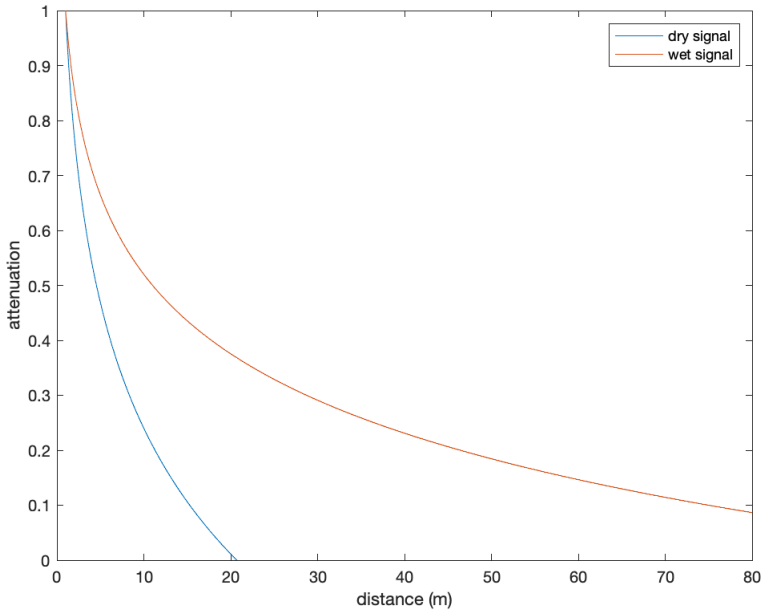


Figure 3.3: Logarithmic attenuation curves for the dry and wet signal of the spatial sound system. The dry signal uses a coefficient of $c = 0.76$ and the wet signal uses $c = 0.48$

is solved by casting visibility rays from the listener to the sound sources using the Möller-Trumbore intersection algorithm.[14] The rays are tested for intersections against a limited selection of meshes with low polygon counts to minimize the computational load. When an intersection is detected in the sound path the direct sound is silenced and the reverb volume is lowered to 20%. For smoother transitions the volume is faded in and out linearly when going between an obstructed and unobstructed sound path.

3.3 Gsound integration

To integrate the sound propagation engine, an interfacing class between the application and Gsound was created. The class handles information flow between scene objects and Gsound objects, see Figure 3.4. When a scene is set up, meshes are loaded in from wavefront files in primary memory and stored in a mesh class inside the application. Gsound uses its own mesh class which is optimized for its acoustic simulations. Mesh data from the application is loaded into a preprocessing function that simplifies the mesh through voxelization and decimation and then generates an edge visibility graph for the edge diffraction algorithm. Only meshes that are manually marked to be part of the acoustic simulation are translated into Gsound mesh objects. The position and orientation of the listener in the Gsound should be the same as the camera in the scene which is also handled by the interfacing class and updated from the main application loop. The same goes for the sound sources which have to be provided with both positioning and audio data from the scene. A sound source class in the application acts as a controller for position and playback of its Gsound counterpart. When loaded with an audio file it is converted into the sound buffer class used in the sound rendering of Gsound.

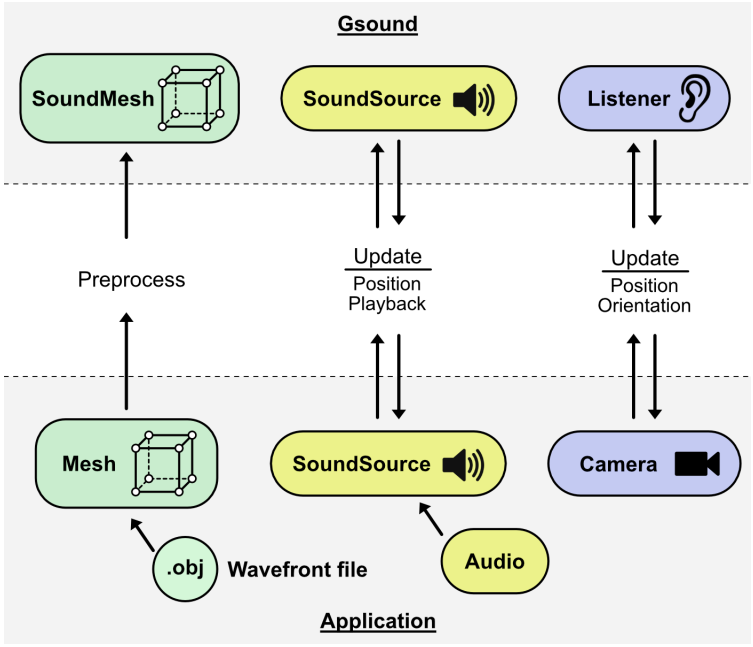


Figure 3.4: The objects in Gsound and the main application that are connected using an interfacing class. Meshes are used as static preprocessed objects in Gsound. The sources and the listener need to be continuously updated with movement and audio data.

3.4 User study

To evaluate how the listening experience of the real-time simulated acoustics with Gsound compares to the more traditional convolution reverb implementation a user study was conducted. The goal of the study was to see if interactive sound propagation enhances the immersion and brings a more realistic sound experience to flat screen interactive experiences with headphones. The sound presented in the study was limited to a single sound source playing an audio recording of a male human voice. The test participant got to evaluate the realism of the acoustics in the sound from various positions in the scene using the two sound renderings methods. To investigate how the different rendering methods fare in different types of environments the test participant gets to interact with three different indoor environments. Two of the environments are open rooms at different scales represented by a large scale cathedral and a smaller scale wine cellar, shown in Figures 3.5 and 3.6. The third environment has the purpose of testing a more segmented location with multiple rooms represented by an apartment, see Figure 3.7 and 3.8.

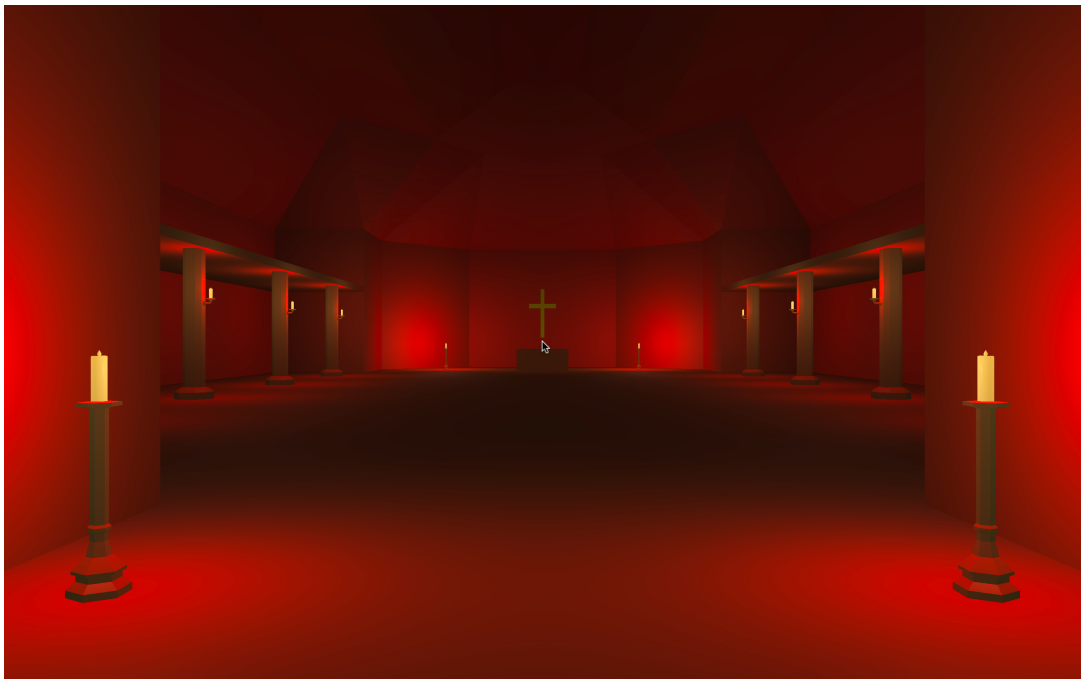


Figure 3.5: The large cathedral environment.

Figure 3.6: The cellar environment.

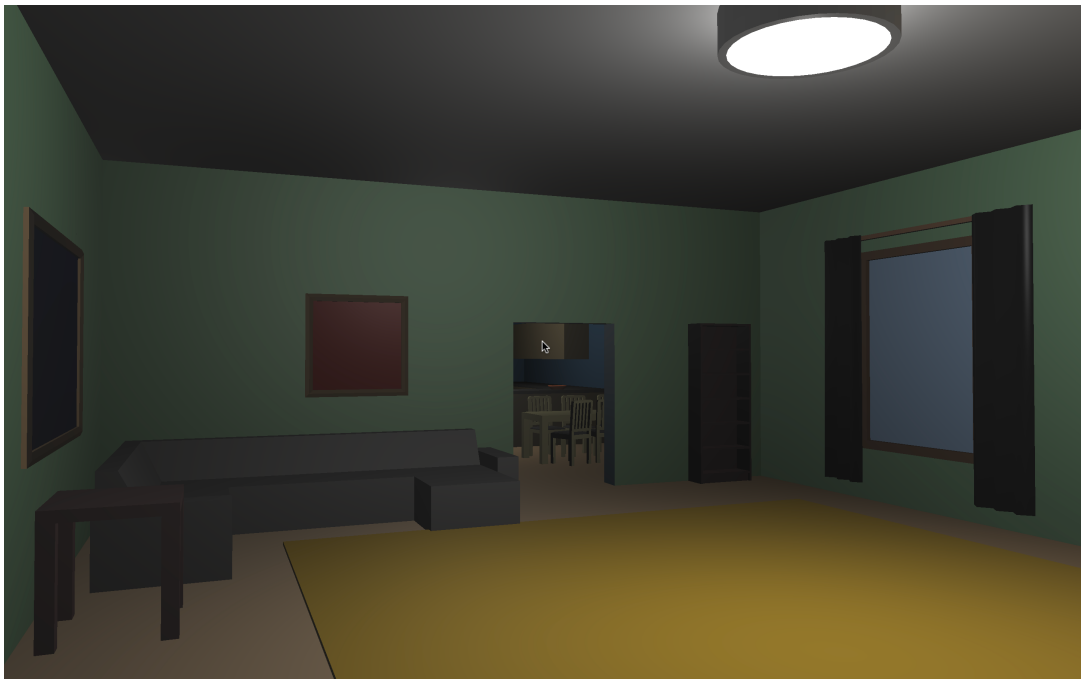


Figure 3.7: The apartment environment.

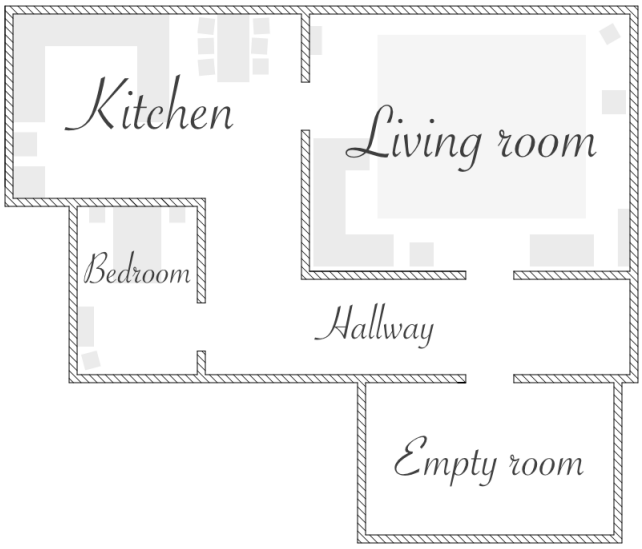


Figure 3.8: The layout of the apartment. The segmented layout creates a more challenging sound rendering environment with walls that block sound coming from other rooms. Differing geometry and materials in the rooms should also result in different acoustics.

The test scenes use abstract graphics but represent familiar environments hinted by its geometry and associative objects. Each rendering method is linked to a number button on the keyboard which enables the test participant to switch between them and compare the sound while interacting. What method is connected to what button is not known to the participant and it changes between the different scenes in the test. This is to avoid bias in the comparison where knowledge of what the rendering methods are could influence the answers. Sound sources are represented by green spheres in the application, see Figure 3.9. The sources are

moving around the scene so that the user participant can experience the sound coming from different positions in the room.

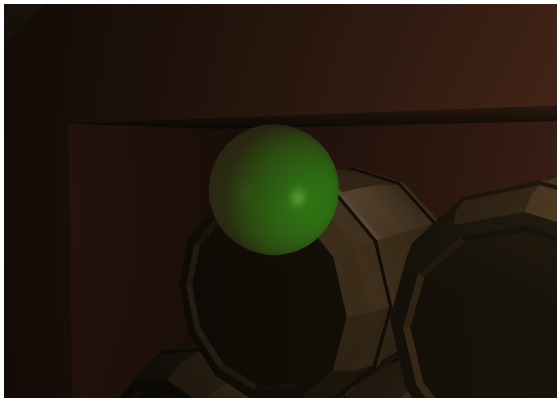


Figure 3.9: The sound source is represented by a green sphere in the test environment.

3.4.1 Test procedure

The test participants were 10 students at Linköping University. They were assumed to have experience with computers and have some understanding of how to navigate through digital 3D space with traditional camera controls using keyboard and mouse as input. The test began with a brief explanation of what the test was about as well as instructions of the controls of the application. The test participant was told to focus on the acoustics of the experience and reflect on the realism and immersion of the different rendering methods. Interaction with the environments was then done in the order shown in Figure 3.10. Between each environment the participant got to answer questions in a questionnaire, see Figure 3.11. The questions were about which rendering method was best as well as perceived properties relating to scale, materials and visual match. The answer options for the perceived properties of the sound were in a 5-point Likert scale that for example goes from small to large. While answering the questions it was possible to go back to the application and listen to the sounds of the two rendering methods again. After each form had been answered the participant got to comment verbally on the two rendering methods in a brief interview relating more generally to realism and immersion. The comments in the interview were written down by the conductor of the user test.

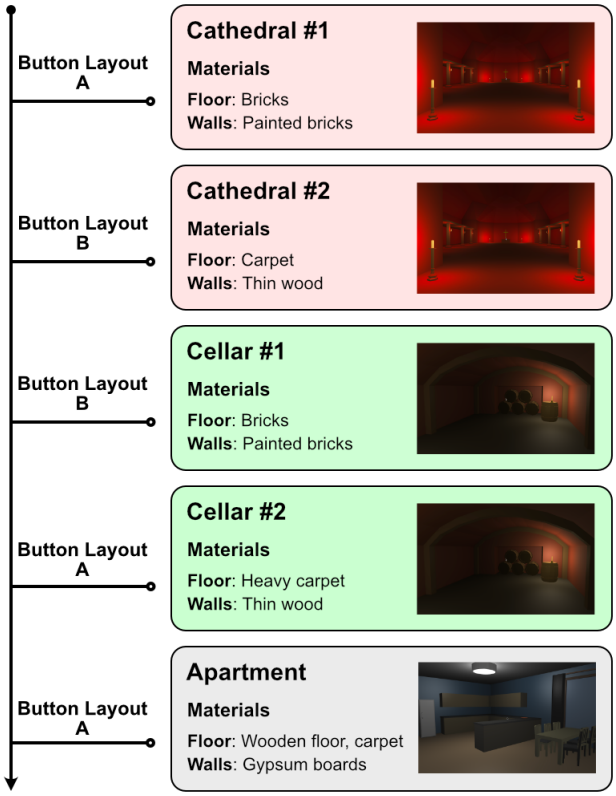


Figure 3.10: The test setup for each environment in the user test. The cathedral and cellar environments are presented twice using different materials. Button layout A and B decides which rendering method is connected to which number button.

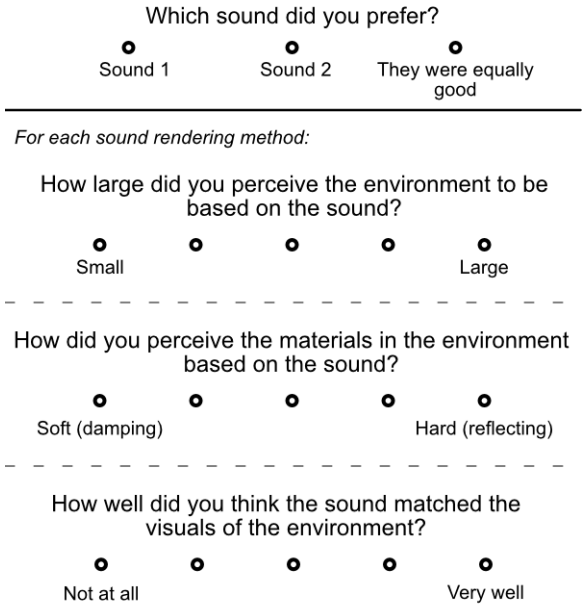


Figure 3.11: The questions in the form that is used for each environment. The participants get to answer which rendering method they thought were best and rate the perceived properties of the two methods.

3.5 Specular sampling based on spatial resolution

The initial sampling directions in ray tracing solutions for sound propagation are usually cast within a uniform distribution to get unbiased directivity. An alternative method would be to sample biased directions based on the spatial resolution of human hearing. This would only make sense for backward ray tracing, where the rays are traced from the listener, and for specular paths which give the most information of the location of a sound source. Using Gsound, specular paths are generated through a hybrid of ray-tracing and image source method. The method generates accurate specular paths between the sources and listener by applying the image source method on triangles gathered from randomized sampling by ray tracing from the listener position. By changing the generation of random directions in the initial sampling phase to be biased in the directions that are perceptually more important for localization of objects the total number of rays could possibly be decreased or focused in certain directions.

Random directions are commonly calculated using spherical coordinates, shown in Figure 3.12. The azimuth θ and inclination angles using a uniform distribution are then calculated as:

$$\begin{aligned} \theta &= \text{rand} \cdot 2\pi \\ \phi &= \arccos(1 - 2 \cdot \text{rand}) \end{aligned} \tag{3.3}$$

and the final vector becomes:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cdot \sin(\phi) \\ \sin(\theta) \cdot \sin(\phi) \\ \cos(\phi) \end{bmatrix} \tag{3.4}$$

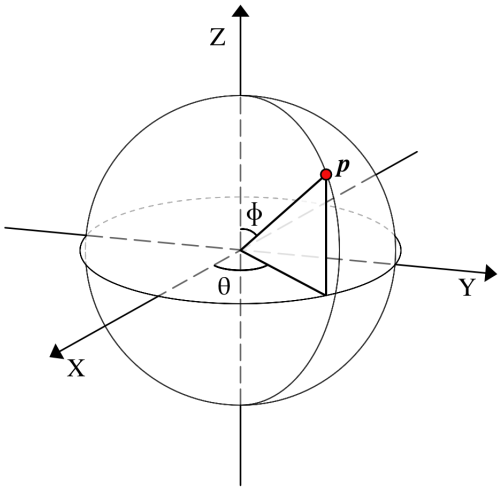


Figure 3.12: A spherical coordinate system. The position of a point p is defined by the azimuth angle θ in the xy-plane and the inclination angle ϕ defined from the z-axis.

To minimize the number of sinusoidal operations, which are computationally heavy, it is possible to avoid the inclination angle ϕ and instead use a scalable vector for the z-component. Random directions sampled from a uniform distribution can then be calculated using two random variables u1 and u2:

$$\begin{aligned} \text{u1} &= 2 \cdot \text{rand} - 1 \\ \text{u2} &= \text{rand} \end{aligned} \tag{3.5}$$

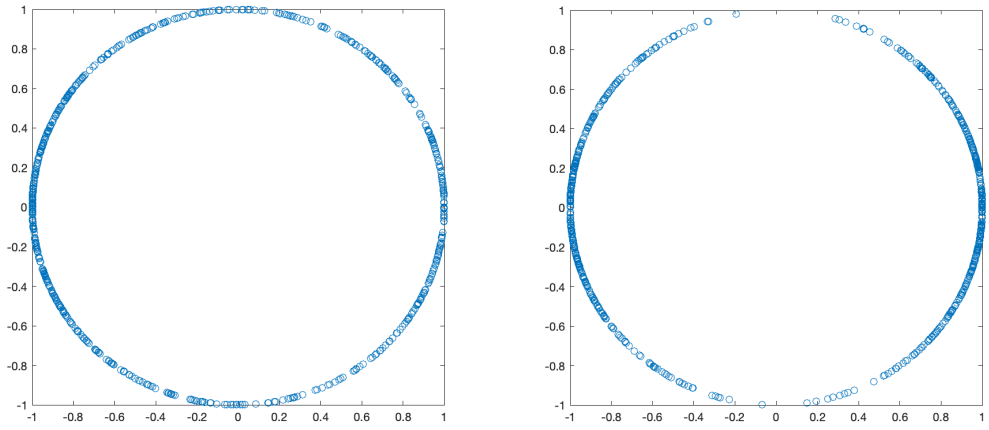
where rand is a random value between 0 and 1. From these variables a radius r and azimuth angle θ can be calculated:

$$\begin{aligned} r &= \sqrt{1 - \mathrm{u1}^2} \\ \theta &= 2\pi \cdot \mathrm{u2} \end{aligned} \tag{3.6}$$

The final direction vector then becomes:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cdot \cos(\theta) \\ r \cdot \sin(\theta) \\ \mathrm{u1} \end{bmatrix} \tag{3.7}$$

To change the random generation to match that of the spatial resolution of human hearing, uniform distributions are replaced by normal distributions. Unlike a uniform distribution, a normal distribution does not generate values between two values. The probability density function is instead defined by the selected standard deviation and variance. To only get values within the defined range the probability density function needs to be truncated which can be done by regenerating the invalid random values. Points generated this way for the horizontal and vertical plane based on the spatial resolution of sound can be seen in Figure 3.13.



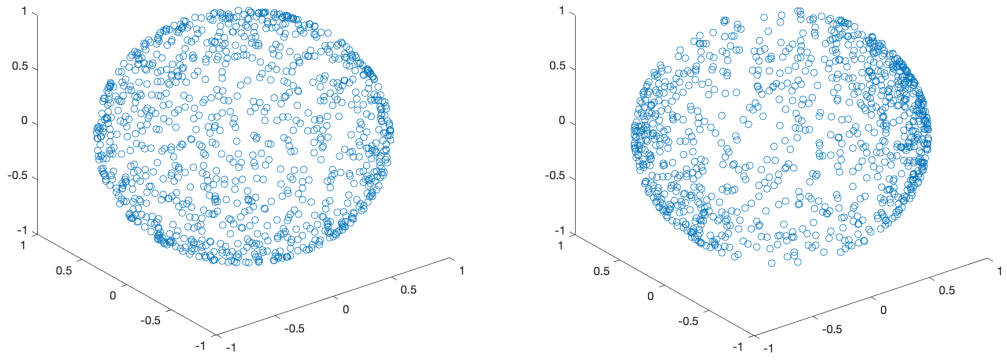
(a) Horizontal plane

(b) Vertical plane

Figure 3.13: Normal distributions generated for the horizontal and vertical plane. The localization of sounds in the horizontal plane is the most accurate in front of the listener and least accurate directly to the right and left. The localisation of sound in the vertical plane is considerably less accurate. Sounds that come directly from above or below the listener are much harder to locate than sources that lie on the horizontal plane.

The change from uniform to normal distributions was implemented into the specular sampling of Gsound using the C++ standard library functions for random generation using the Mersenne Twister 19937 random generator. Floating point numbers are generated within a normal distribution provided the mean value and the standard deviation. The random values in the algorithm for spherical points should fall between zero and one with a mean of 0.5. The selected standard deviations σ associated with the horizontal and vertical planes selected as $\sigma = 0.12$ and $\sigma = 0.25$ respectively to approximate a probability density representing the spatial resolution for sound. The resulting spherical points compared to uniform sampled ones are shown in Figure 3.14.

A difference between an unbiased uniform sampling and a biased sampling method based on localization accuracy in different directions is that the latter uses a coordinate system defined relative to a human head. This means that a direction vector generated from the points



(a) Uniform distribution

(b) Normal distribution

Figure 3.14: 1000 points generated using a uniform distribution and the proposed normal distributed method.

in the sphere needs to be rotated with the listener orientation in the scene. This is done by applying the orientation matrix of the camera, representing the listener in the application, to every randomly generated vector.

4 Results

The results from the user study comes from the answers of the questionnaire as well as notes from interviewing the participants during the test. The test participants did not know if the sounds they were listening to in the test were the simulated acoustics with Gsound or the custom stereo solution. Each rendering method was instead labeled as “sound 1” or “sound 2” which alternated between the different interactive environments.

4.1 Church environment

The church environment, representing a large open indoor area, was used two times in the user study with different material setups. The first church used harder acoustic materials with brick floors and walls. The results from the questionnaire are shown in Figure 4.1.

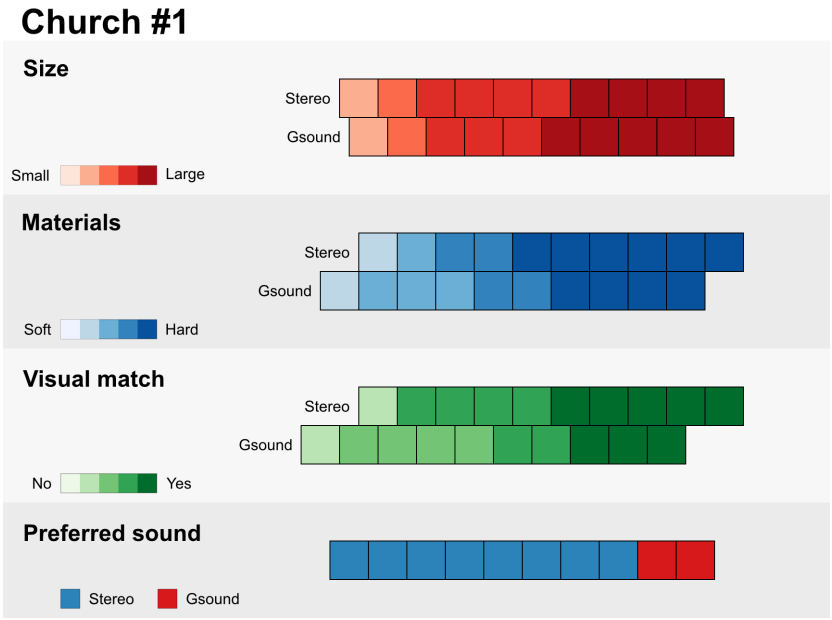


Figure 4.1: The results from the questionnaire for the first church environment with harder acoustic materials.

The majority of the participants thought that the stereo sound was better and sounded more realistic compared to the real time simulation. The stereo sound was perceived by some to have more reverb and was easier to locate the sound source. A couple participants thought Gsound sounded more realistic with how the reverb was dampened at different positions.

The second church had softer acoustic materials with a carpet floor and thin wood walls. The answers from the questionnaire can be seen in Figure4.2.

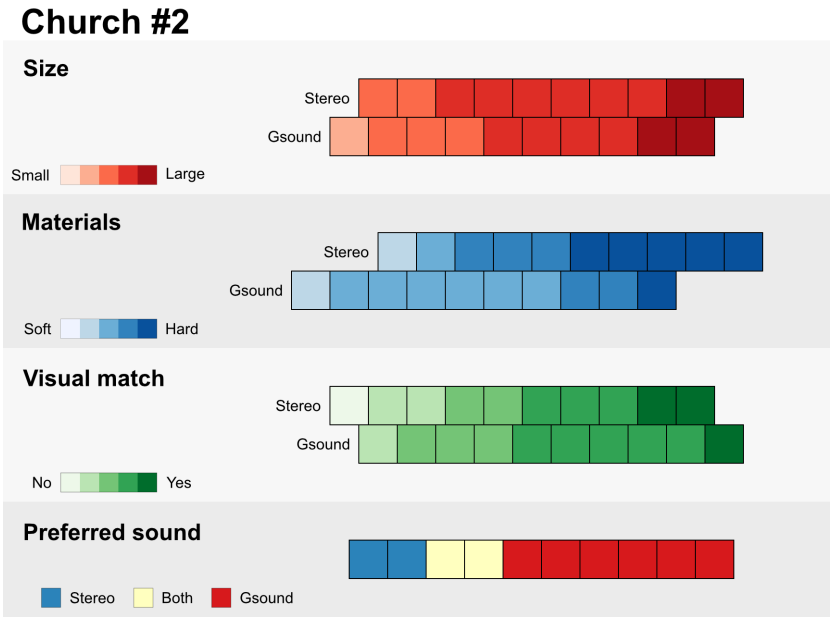


Figure 4.2: The results from the questionnaire for the second church environment with softer acoustic materials.

In the second church the preference were more split and many thought the two rendering methods sounded quite similar in most aspects. Some participants thought the reverb of the

stereo version sounded a little strange and that it sounded more like a direct echo than correct reverberation. Gsound were perceived as more realistic close up to the audio source, where the stereo sound had too much reverb.

4.2 Cellar environment

To test a smaller open environment the test participants got to interact with a cellar environment, which similar to the church was tested two times with different material setups. The questionnaire response for the first cellar, which used harder acoustic materials, is shown in Figure 4.3.

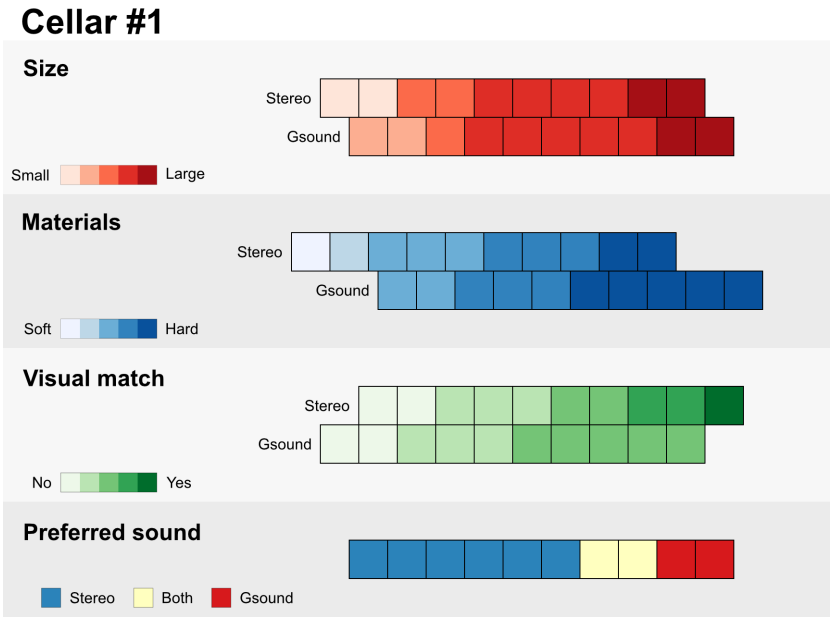


Figure 4.3: The results from the questionnaire for the first cellar environment with harder acoustic materials.

Overall most thought both sound methods had too much reverb and did not match what they expected from the visuals of the cellar. The acoustics instead gave them the impression of a much larger place. Many thought that the sound produced by Gsound had especially strong reflections and that it was harder to localize the sound source. One person experienced a weird sound from the stereo solution with a buzzing sound effect.

The second cellar had a test setup using softer acoustic materials. The results from the questionnaire are presented in Figure 4.4.

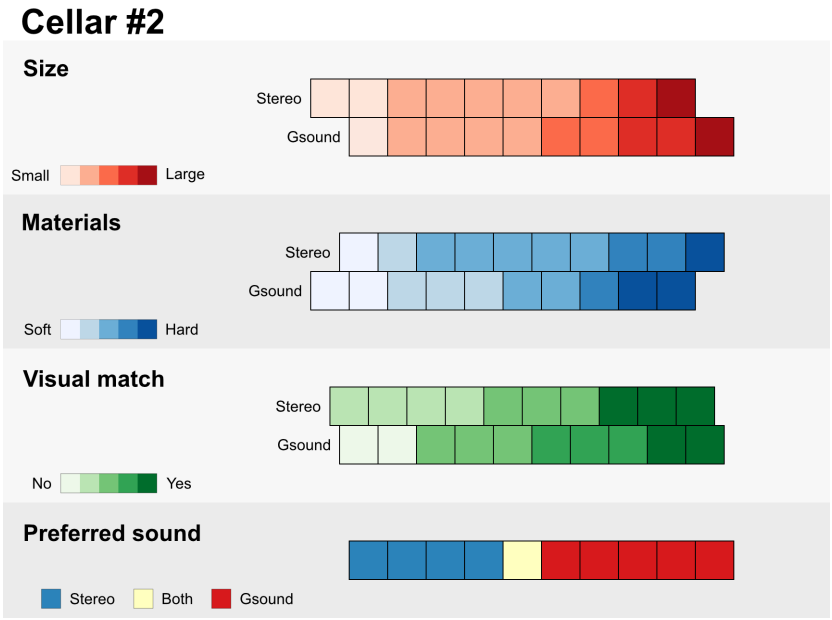


Figure 4.4: The results from the questionnaire for the second cellar environment with softer acoustic materials.

Compared to the last cellar test, the participants considered the acoustics to better match their expectations of how the cellar should sound. When it comes to which sound they preferred it was very split with many conflicting answers. Some participants thought the stereo version had too much echo and others that Gsound had too much echo. Almost everyone preferred the sound they perceived to be less reflective. In the stereo sound multiple people perceived a strange base resonance. Some people also noticed a delay effect in the stereo sound which one participant described sounding almost like a flanger effect. Localization of the sound source was again thought to be easier using the stereo solution compared to the real-time simulation i Gsound.

4.3 Apartment

The last test environment was a large apartment with multiple rooms branching out from a hallway. The purpose of the apartment was to test a more segmented environment where the sound source can be in a different room than the listener. Results from the questionnaire for the apartment are shown in Figure 4.5.

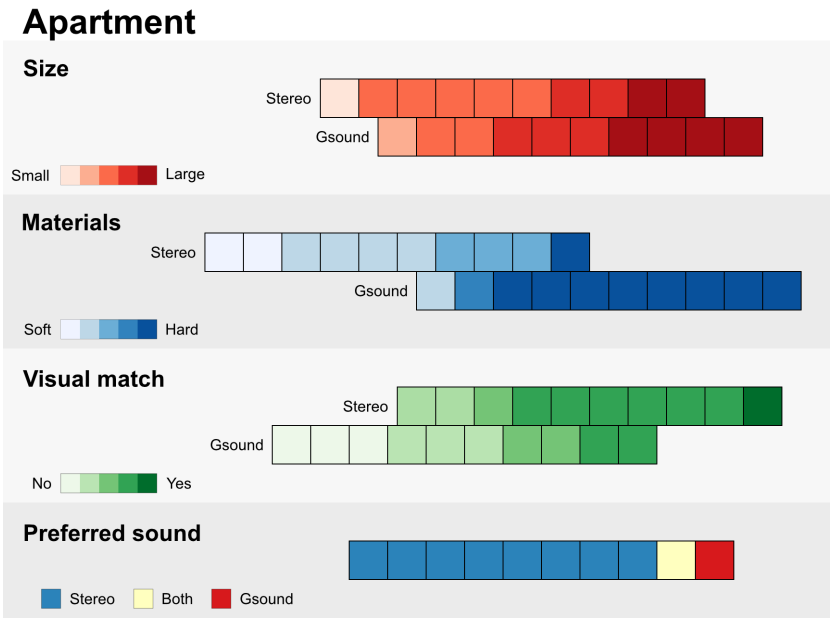


Figure 4.5: The results from the questionnaire for the apartment environment.

In this test there was a huge difference in the perception of the two sound rendering methods. Almost everyone thought both sounds had way too much reverb for a furnished apartment. Most did however experience a much more reflective sound with Gsound compared to the stereo solution which they thought matched the visuals slightly better. When it comes to sound from a source in another room, most thought the simulated sound with reflections and diffraction sounded unrealistic and thought that sound came through the walls. The apartment with Gsound was described as sounding like one big room, without the walls in between. Instead most people instead preferred the sound obstruction solution in the stereo sound, but thought its transition in the sound when the source came in and out of view was too drastic, which was unrealistic. A couple of the test participants did however prefer the simulated audio from other rooms and thought that it was more physically correct and realistic.

4.4 Specular sampling based on spatial resolution

The computation time of the normal distributed sampling method based on the spatial resolution of human hearing was timed and compared to Gsound, which uses its own implementation of a random generator. A version replacing the Gsound random generator with a C++ standard library uniform distributed generator was also timed and compared. Both the normal distributed version and the C++ uniform version uses the Mersenne Twister 19937 random generator. The averaged computation times for the three algorithms are shown in Figure 4.6. From these results it can be seen that the C++ uniform sampling is 1.7 times slower than the random implementation in Gsound. The normal distributed version is 3.6 times slower than the Gsound implementation and 2.1 times slower than the C++ uniform sampling. In the limited testing done by the author within the same test environments of the user study there were no major perceptive differences between the two sampling methods.

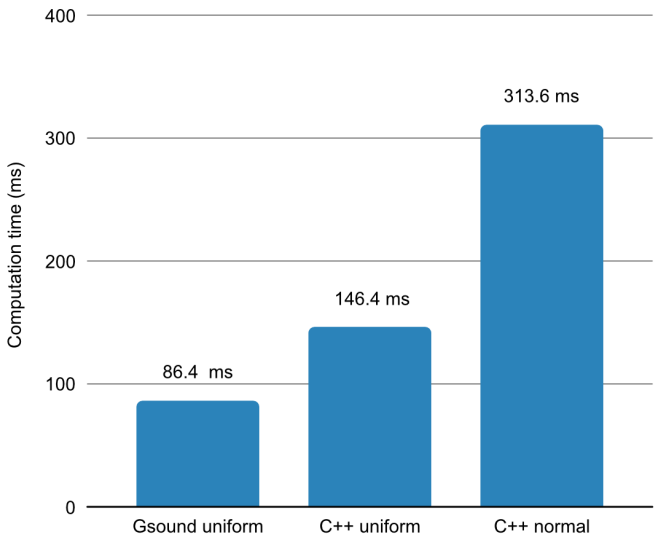


Figure 4.6: The time it takes to generate 1 000 000 random directions using three different randomization methods.

5 Discussion

In the work of this thesis the experience of acoustics in interactive experiences was explored as well as the potential use of a different sampling method based on localization accuracy in hearing. The user study covered a broad number of experience qualities and generated many interesting results. The computation time of the specular sampling method was measured but the listening is yet to be tested thoroughly.

5.1 Simulated acoustics for flat screen experiences

In the user study the aim was to explore how simulated acoustics affects immersion and realism in interactive experiences with graphics on a regular flat screen. The main point of interest was to see if there are any benefits of updating the impulse response in real-time compared to using a pre-calculated general impulse response for the environment. Since the relevant questions mostly deal with the subjective experience of the listener, the test consisted of a combination of quantitative and qualitative methods in the form of a questionnaire and an interview. The points of measurement in the questionnaire were size, materials and visual match, which were measured using 5-point Likert scales. Since the reference labels in the scales are context dependent and subjective, the collected data should not be interpreted in other contexts outside of this study. Instead the answers in the questionnaire show the differences between the two sounds more clearly and work as a complement to the interview about the listening experience.

5.1.1 Participant selection

The selection of participants in the user study were limited to 10 people, all of which were students at Linköping University and most studying a technical program. The small sample size makes it hard to conclude anything about any quantitative measurements but should be enough to get an understanding of the range of experiences with the two rendering solutions and compare them. A majority of the participants had some previous experience with interactive experiences like video games which might not have been the case with a different selection of people. This is a factor that might benefit the ease of interaction with the application, but it could also alter expectations of the acoustics.

5.1.2 Acoustic experience

The results from the user study show a tendency of a preference for the custom stereo solution for the environments that used more reflective acoustic materials while there was more of a preference for Gsound in the environments with more dampening acoustic materials. One possible explanation for this could be the harder reflections that were experienced with Gsound, which is most likely the reflections coming directly from a nearby wall. These reflections could be considered strange. The impulse response used for the static stereo rendering was calculated with the same simulation for one position where the sound source and listener were positioned at a central location in the scene with equal distances to the walls. This was done to avoid an unbalanced stereo image for an impulse response that is supposed to be used for every spot of the environment. In other words, without the real-time simulation you lose the hard reflections that can be heard when standing closer to a wall.

A majority of the participants consistently expressed that it was easier to localize the sound source with the stereo sound. This is most likely due to how the stereo solution uses basic panning for directionality while Gsound uses the direction data of every reflection reaching the listener to generate the final sound. This makes it so that the sound not only comes from the direction of the sound source, but also nearby reflective surfaces, making localization of the sound more difficult. Another thing that affects localization with Gsound is that there seems to be a slight delay before the directionality is updated when reorienting the camera view. The way the sound rendering is set up forces an update of the sound propagating data when updating the listener orientation which causes a noticeable delay if the simulation is not fast enough.

Interestingly, there were a few participants that explicitly commented on how the reverberation changed at different positions in the room with Gsound. It is possible that people are less likely to notice more physically correct acoustics in audio when they are not fully immersed visually. Expectations of how things should sound are largely based on previous experiences which could be different for media shown on a flat screen. For example video games and many movies rarely have physically correct acoustics. To investigate if visual immersion factors change acoustic expectations, further tests should compare interactive sound propagation for environments presented on a flat screen with virtual reality. Another thing that has an effect on immersion is the input method for interaction. Controlling a virtual camera with keyboard and mouse is not immediately intuitive for people unfamiliar with that method of interaction. Participants who maybe did not have this experience had one extra thing to think about and might not have walked and looked around as much in the test. This could have had some effect on the results. Higher immersion might lead to other expectations of how things should sound and more movement would likely make the effects from Gsound more clear.

There was no major difference in the perceived size of the environments between the two rendering methods. This is a logical outcome since the reverb of both rendering methods are generated using the same simulation software with the same acoustic parameters. Still, the generalized impulse response of the stereo sound could be expected to give a different sense of scale but that does not seem to be the case based on the results of this study. There could however be other environments, not tested in this user study, with more variation in the scale and geometry that would make the use of a single impulse response sound strange for the static stereo solution.

The metric of visual match was supposed to indicate how well the participants thought the acoustics matched what was presented on screen. This could be related to the size and geometry of the room, but there are also reference objects in the scene that hints what type of room it is. There were no big differences between the two rendering methods related to this except for the apartment where the reflections from other rooms greatly affected the result. Since both sounds use the same simulation to generate the reverb, the small differences are not a completely unexpected result. It is unclear how different visual properties set the

expectations of how things should sound. It would be interesting to test the same sound, but change the materials in the graphics, with different textures and lighting, and see if that has any effect on how we experience acoustics.

5.1.3 Multiple rooms

The most complex environment was the apartment which consisted of multiple rooms connected to a hallway, which meant that the sound source and the listener could be separated by a wall. This is a challenging environment to recreate acoustically since early reflections and diffraction become much more noticeable with enclosing walls and door openings. Reflections and diffracted sound waves reaching the listener from a door opening to another room can not be replicated in traditional methods but is possible using a geometrical acoustic simulation. This makes a segmented environment like the apartment a case where there should be a big difference between the two rendering methods. To approximate some type of sound transmission from the walls or reflections reaching the listener from another room in the stereo solution the sound was limited to a lower volume reverb signal if the source was not visible. It is also hard to do a realistic transition between when the sound source goes from visible to not visible. In the test, these transitions were done by fading the volume in and out, taking circa one second. Another difference in the case where the source visibility is obstructed by a separating wall is sound directionality. In the simulation the sound comes from the direction of the door opening or a reflective wall while in the stereo solution the sound still comes from the direction of the source position. Interestingly, most participants in the study preferred the sound blocking effect from the stereo solution over the much more physically correct simulation, thinking that it sounded like there was just a big room without walls in between. Few people realized that the reflections were coming from the door openings or bouncing around corners with Gsound. They did however think that the transitions between a visible and obstructed source were more natural with Gsound. This indicates again that there might be different expectations of how things should sound for media presented on a flat screen and that simulating acoustic effects for these situations might not be worth it. At the same time a couple participants noticed the difference and thought the effects produced with Gsound were much more realistic. It is hard to say from this study which has a relatively small sample size how this split of opinions is distributed or what made some perceive the simulated audio as unrealistic.

One explanation to some of the peculiarities of the answers for the apartment test could be the chosen reference environment. The expectations of an apartment with furniture is that there should not be any major echoes. A mistake that was done with the setup of the apartment layout for the test was that it contained way too much open space without dampening furniture. Even though the materials for the walls and ceiling were gypsum boards and the floor was carpet, it was not enough to dampen the reflections in a simulation. Smaller rooms with dampening furniture would have less echoes and present a more familiar acoustic environment for an apartment. Moreover, the hallway between the rooms was completely empty making the sound echo through the entire apartment which was one thing people commented on as being unrealistic for an apartment. To get a better understanding of how reflections and diffraction affect immersion and realism in segmented environments more tests should be done with different types of environments and different visuals.

5.1.4 Future work

There are some things that were left out of this user study that could be interesting to investigate. One is the use of head related transfer functions (HRTFs) which adds the filtering effect of sound interacting with the head and ears which depends on the incident angle of the sound. This would give a more natural sounding result with better localization of sound, which could be especially realistic in a combination with simulated acoustics. It does how-

ever also add complexity in both implementation and test setup. A HRTF is in theory completely individual and would need to be measured for every user to get a correct experience, which is not feasible. In a user study there would instead have to be a setup phase where an HRTF that is close enough to the user is selected from a library of HRTFs. Variability in the matching of HRTFs would create a slightly different listening experience for every participant that could affect the results.

Another thing that could be investigated is the experience with multiple sound sources and more variety in the types of sound. In the user study the test environments were limited to a single sound source with a human voice as playback. It is possible that a higher amount of sound sources and different audio content could be perceived differently using a simulation compared to pre-applied reverb.

One thing that could have been improved in this user study is the custom implementation of spatial sound. The strange reverb sound experienced by some in the less reflective test environments with the stereo sound can most likely be explained by the simplified reverb implementation used in the test. The stereo sound was built up using two seperate audio signals, one containing the original recording, representing the dry signal, and the other containing the pre-applied wet reverb signal. While this provided an easy way to mix the audio for the test, the unintended consequence was that if the two signals are similar and close enough in time, interference will create a comb filter effect. This could be mitigated with an implementation where the reverb is applied in real-time using a fast Fourier transform algorithm, which would only use one signal.

Considering how close the experience of the two rendering methods were in most of the test environments, it might be enough in most cases to use a static reverb from a simulated impulse response to get immersive acoustics. Having a fast acoustic simulator built into a game engine to generate impulse responses for the scenes, by only providing meshes and selecting acoustic materials, could be very useful and greatly improve development workflow. While the dynamic impulse response of an interactive sound propagation engine would provide more physically accurate directional sound with reflections and can simulate diffraction effects, it might not be needed for immersion as indicated by the results of this user study. There would, however, still be the problem of larger environments with variation in size, geometry and materials which could not be covered using a single impulse response. For such a case reverb zones could be implemented to gradually switch the impulse response for a different location using interpolation. This could for example be used in the apartment environment which would make a huge difference if there were more acoustic variance between the rooms. It could also be used for for some larger rooms where reverb zones can be used to interpolate between multiple impulse responses pre-calculated at different positions of the same room. It would be interesting to investigate how acoustics rendered using reverb zones compare to sound rendered with interactive sound propagation.

5.2 Specular sampling

Another thing that was explored was a potential utilization of the property of spatial resolution of human hearing. The idea was to map this property to a probability function that can be used in the sampling stage of backward ray tracing algorithms for acoustics where rays are cast in random direction from the listener position. By changing the sample density in different directions, higher accuracy could be achieved in the directions a human is better at localizing objects in. This method would only make sense for propagation paths that convey accurate directional information about the source to the listener. Because of this it makes more sense to apply it for early specular reflections rather than late diffuse reflections where sound has been scattered in random directions. For ideally specular reflections the source is perceived to be located at the mirror image position from the reflection surface. In

this case the sampling method was applied to Gsound which uses a hybrid ray-tracing and image source method for specular reflections.

The sampling directions were approximated with a probability function using a combination of normal distributions. The average computation time for the function was measured to be about 3.6 times slower than the function used in Gsound. This is a lot slower, but would be negligible if it would make it possible to cast a lower amount of rays which is a task that takes significantly longer time to compute. It is also possible that there is a more efficient random generator implementation that can be used. This is illustrated by using the C++ standard library implementation of the Mersenne Twister 19937 algorithm for a uniform sampling which also is 1.7 times slower on average compared to the Gsound implementation. Handling values falling out of range was done by regenerating the value completely will cause some variance in computation time. The time variance will depend on how likely it is that a value falls out of range which is determined by the selected standard deviations in the normal distributions.

It is unclear from the testing done with the sampling if it is even a valid method for image source methods. The reason why there was no difference detected in the test environments is because they used low polygon assets for the geometry. With the image source based algorithm used in Gsound most specular paths would be found even with a very low amount of specular samples. Every triangle around the listener is most likely detected, forming every valid listener image and diffraction edge. To test the sampling method with the specular algorithm used in Gsound, a scene with much more complex geometry would need to be used. It is still unknown if sampling that is not uniform would create an inconsistent or unbalanced listening experience. However it is possible that a specular cache that stores old specular paths could help with consistency when reorienting the camera which would also move the orientation of the sampling density. It would also be interesting to investigate if the sampling method could be used in a statistical algorithm with ray casting from the listener like a bidirectional path tracer.

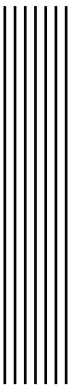
6 Conclusion

The aim of the work in this thesis was to investigate how acoustics simulated in real-time affect the experience of interactive applications. To investigate this, a user study was conducted comparing sound rendered with a sound propagation engine with a more traditional stereo method using a precomputed impulse response. In the study the participants got to interact with different environments and reflect on different experiential qualities of the acoustics and choose which sound they thought was most immersive.

While there were differences between how the two methods were perceived in the user study, it varied depending on the environment and acoustic materials that were used. In open rooms there were overall only relatively small differences in the sound experience and there was even a slight preference for the static reverb method. The biggest difference was for sound coming from obstructed sources where diffraction and early reflections become important. These effects were however not necessarily perceived as being more realistic though, based on the results from this user study. There was no obvious benefit to interactive sound propagation found in this study that would justify the computational cost for an application outside of virtual reality, but it is possible that other environments than was tested, where sound gets obstructed from view, could have some value from simulated reflections and diffraction. Some of the benefits interactive sound propagation with a dynamic impulse response can bring to the sound experience is the accurate dampening and reflectiveness at different positions which was noticed by some participants. At the same time, many participants also thought that it was harder to locate sound sources with the sound propagation engine compared to the stereo solution used in this study which affected immersion.

A separate aim was to explore how the specular resolution for human localization of sound sources could be used to optimize acoustic algorithms. A new sampling method was proposed for backward ray tracing algorithms that replaces uniformly distributed random directions with normal distributions based on spatial resolution. Computing these random directions takes longer time but it would be negligible if it makes it possible to cast fewer rays in the directions with low spatial resolution. In testing the implementation of this with specular reflections in Gsound there were no discernible differences in the sound. To evaluate if the sampling method is feasible for the algorithm used by Gsound, the scene would have to have very dense geometry. It could also be tried with different backward ray tracing algorithms than the hybrid image-source method used in this test.

Overall, sound from precomputed impulse responses were perceived as sounding very similar to a real-time simulation. Having fast and simple to use acoustic simulations available to generate custom impulse responses could be a valuable feature to have in for example a game engine. Using acoustic simulations to generate the impulse responses beforehand is likely enough to create convincing sound for most interactive experiences outside of virtual reality.



Bibliography

[1] Jont Allen and David Berkley. "Image method for efficiently simulating small-room acoustics". In: *The Journal of the Acoustical Society of America* 65 (Apr. 1979), pp. 943–950.

[2] M. A. Biot and I. Tolstoy. "Formulation of Wave Propagation in Infinite Media by Normal Coordinates with an Application to Diffraction". In: *The Journal of the Acoustical Society of America* 29.3 (1957), pp. 381–391.

[3] Jens Blauert. *Spatial Hearing: The Psychophysics of Human Sound Localization*. The MIT Press, Oct. 1996.

[4] Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou. "Interactive sound propagation with bidirectional path tracing". In: *ACM Transactions on Graphics* 35 (Nov. 2016), pp. 1–11.

[5] C. R. A. Chaitanya, John Snyder, Keith Godin, Derek Nowrouzezahrai, and Nikunj Raghuvanshi. "Adaptive Sampling For Sound Propagation". In: *IEEE Transactions on Visualization and Computer Graphics* 25.5 (May 2019), pp. 1846–1854.

[6] Claus Lynge Christensen. "A new scattering method that combines roughness and diffraction effects". In: *The Journal of the Acoustical Society of America* 117.4 (2005), pp. 2499–2499.

[7] Thomas A. Funkhouser, Ingrid Carlbom, Gary W. Elko, Gopal Sarma Pingali, Mohan Sondhi, and Jim West. "A beam tracing approach to acoustic modeling for interactive virtual environments". In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), pp. 21–32.

[8] Andrew Kelly. *libsoundio*. Version 2.0.0. URL: `http://libsound.io`.

[9] Mendel Kleiner, Bengt-Inge Dalenbäck, and Peter Svensson. "Auralization-An Overview". In: *Journal of the Audio Engineering Society* 41.11 (Nov. 1993), pp. 861–875.

[10] R.G. Kouyoumjian and P.H. Pathak. "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface". In: *Proceedings of the IEEE* 62.11 (1974), pp. 1448–1461.

[11] A. Krokstad, S. Strom, and S. Sørsdal. "Calculating the acoustical room response by the use of a ray tracing technique". In: *Journal of Sound and Vibration* 8.1 (1968), pp. 118–125.

[12] H. Kuttruff. "A simple iteration scheme for the computation of decay constants in enclosures with diffusely reflecting boundaries". In: *The Journal of The Acoustical Society of America* 98 (July 1995), pp. 288–293.

[13] H. Medwin. “Shadowing by finite noise barriers”. In: *The Journal of the Acoustical Society of America* 69.4 (1981), pp. 1060–1064.

[14] Tomas Möller and Ben Trumbore. “Fast, Minimum Storage Ray-Triangle Intersection”. In: *J. Graph. Tools* 2.1 (Oct. 1997), pp. 21–28.

[15] Gregor Mückl and Carsten Dachsbacher. “Precomputing Sound Scattering for Structured Surfaces”. In: *Eurographics Symposium on Parallel Graphics and Visualization*. Ed. by Margarita Amor and Markus Hadwiger. The Eurographics Association, 2014.

[16] Ken Pohlmann and F. Alton Everest. *Master Handbook of Acoustics*. Fifth Edition. McGraw-Hill/TAB Electronics, 2009.

[17] Matthew Rosen, Keith Godin, and Nikunj Raghuvanshi. “Interactive sound propagation for dynamic scenes using 2D wave simulation”. In: *Computer Graphics Forum* 39 (Dec. 2020), pp. 39–46.

[18] Lauri Savioja and U. Peter Svensson. “Overview of geometrical room acoustic modeling techniques”. In: *The Journal of the Acoustical Society of America* 138.2 (2015), pp. 708–730.

[19] Carl Schissler and Dinesh Manocha. “Adaptive Impulse Response Modeling for Interactive Sound Propagation”. In: *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. I3D ’16. Redmond, Washington: Association for Computing Machinery, 2016, pp. 71–78.

[20] Carl Schissler and Dinesh Manocha. “GSound: Interactive Sound Propagation for Games”. In: *Proceedings of the AES International Conference* (Feb. 2011).

[21] Carl Schissler and Dinesh Manocha. “Interactive Sound Propagation and Rendering for Large Multi-Source Scenes”. In: *ACM Transactions on Graphics (TOG)* 36 (2016), pp. 1–12.

[22] Carl Schissler, Ravish Mehra, and Dinesh Manocha. “High-Order Diffraction and Diffuse Reflections for Interactive Sound Propagation in Large Environments”. In: *ACM Transactions on Graphics* 33 (July 2014), pp. 1–12.

[23] Dirk Schröder. “Physically Based Real-Time Auralization of Interactive Virtual Environments”. PhD thesis. Jan. 2011.

[24] Uwe Stephenson. “An Energetic Approach for the Simulation of Diffraction within Ray Tracing Based on the Uncertainty Relation”. In: *Acta Acustica united with Acustica* 96 (May 2010), pp. 516–535.

[25] U. Peter Svensson, Roger I. Fred, and John Vanderkooy. “An analytic secondary source model of edge diffraction impulse responses”. In: *The Journal of the Acoustical Society of America* 106 (1999), pp. 2331–2344.

[26] Nicolas Tsingos, Thomas Funkhouser, Addy Ngan, and Ingrid Carlbom. “Modeling Acoustics in Virtual Environments Using the Uniform Theory of Diffraction”. In: (June 2001).

[27] Michael Vorländer. “Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm”. In: *The Journal of the Acoustical Society of America* 86.1 (1989), pp. 172–178.