

Acoustic Mapping of Simulated Environments

Jacob Daskocil
doskoj@rpi.edu
Rensselaer Polytechnic Institute
Troy, New York

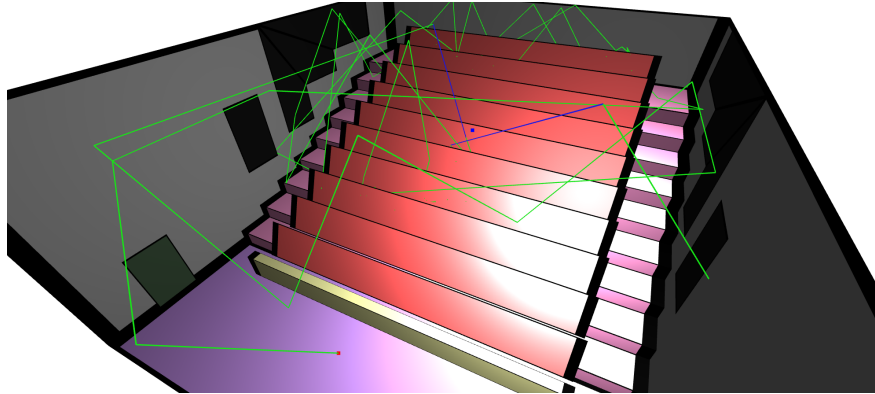


Figure 1: Auditorium model showing a single pathtraced ray.

ABSTRACT

We present a program for rendering impulse responses for simulated environments using raytracing methods, and for using those impulse responses to render a system response for input audio.

1 INTRODUCTION

Acoustic mapping is an area with numerous applications, for architects and for artists. Physically, sound is a wave that travels through an environment, and reflects off of objects. It shares many properties with light, including how specular reflections and diffusion are effected by the materials with which it comes into contact. Therefore, it is possible to simulate sound using techniques used for simulating light, such as raytracing.

In raytracing for image files, each sample contains measurements for three wavelengths, corresponding to red, green, and blue colors. For this project, we will use a similar method and generate output for six different frequencies, representing octaves of 125 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, and 4000 Hz. These values have been chosen due to the availability of material property data from other sources.

This project focuses on generating impulse responses for a simulated environment, and then on using those impulse responses to simulate audio output.

2 THE IMPULSE RESPONSE

The impulse response of a system is the output given when the input of the system is an impulse. This project is implemented digitally, and so all signals involved will be in discrete time. For the purposes of room acoustics, the impulse response represents the

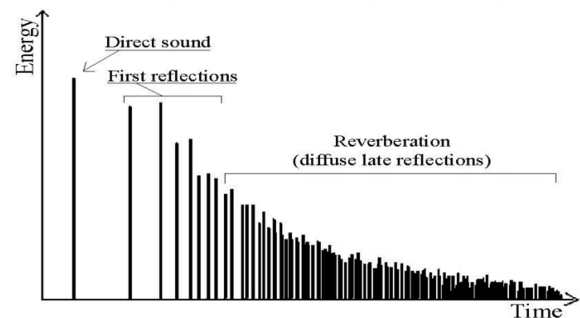


Figure 2: A sample impulse response

sound heard from a listening point when an impulse, defined by Eqn 1 is generated at a source point.

$$\delta[t] = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{else} \end{cases} \quad (1)$$

In an environment where the energy reaching a listening point is a result of either direct sound or from sound waves reflecting off of scene geometry, the impulse response of a system will consist of a series of impulses, scaled by the energy contained by a wave taking that path, and delayed by the travel time of a wave taking that path. Typically there are three main parts to an impulse response shown in Fig 2 [4].

The earliest impulse is the result of direct sound waves from the source to the listener, if there is a direct path available. Following the direct sound will come early reflections. Early reflections are the specular reflections that originate from sound bouncing off of reflective objects that are stronger and clearer than most other

reflections. Following the early reflections is the remainder of the response, the reverberations and late diffuse reflections. This section is the result of low energy sound diffusion, as well as later and weaker specular reflections.

The impulse response is useful because it provides a model for the most basic interaction with a system, and can be used to create any output signal for a system. Given an impulse response of a system $H[t]$ the result of the impulse $\delta[t]$, the result of an arbitrary signal $S[t]$ can be modeled through convolution, using Eqn 2.

$$(S * H)[t] = \sum_{k=-\infty}^{\infty} S[k]H[t - k] \quad (2)$$

3 ACOUSTIC MATERIALS

Material properties play a key role in the propagation of sound in an environment. There are two types of materials that are used in these simulations: travel media and reflection materials.

3.1 Travel Media

The travel medium is a material through which sound travels. The travel medium has two major effects on sound traveling through it. It attenuates the energy of sound as it travels, and determines the propagation velocity of sound. Both of these material properties are determined by the molecular structure of the material, gases tend to have stronger attenuation and lower propagation velocities than solids or liquids [3]. The attenuation of energy is dependant on the frequency of the energy. Each travel medium has an attenuation constant α stored as a Vec6f for each of the measured frequencies, where attenuation tends to increase with frequency [4]. The energy decays exponentially as the sound travels through the medium. Using attenuation constants in dB/m α_f for a frequency f , the resulting energy E from initial energy E_0 after traveling a distance d can be calculated using Eqn 3.

$$E = E_0 10^{d\alpha_f/20} \quad (3)$$

Attenuation constants for air are calculated using [5], with a relative humidity of 50% and a temperature of 20 °C. The materials used are shown in Table 1.

3.2 Reflection Materials

The other type of material used in simulations are the reflection materials. These are materials that are assigned to faces in the scene and determine how sound reflecting off the face will behave. There are two properties that are focused on in this method. The first is a scattering constant σ which describes the roughness of a material, how likely it is to have a diffuse reflection rather than a specular reflection. The other property is the absorption of the material. A frequency based absorption constant β describes how much energy at that frequency is absorbed by the material upon an intersection. The new energy E' after a reflection can be calculated by Eqn 4, with absorption constant β and initial energy E .

$$E' = \beta E \quad (4)$$

Reflection material constants are taken from [1] and [4], shown in Table 2.

4 METHODS AND IMPLEMENTATION

This project consists of two main programs, an offline program that uses 3D geometry to render impulse responses, and a real-time program that uses the pre-rendered impulse responses and input audio to render simulated audio in the simulated environment. The impulse response rendering program is written in C++, starting the OpenGL framework and file loading software from Homework 3. The audio rendering program is written in Pure Data, an interactive visual programming language.

4.1 Writing Output to Audio Files

For the rendering of images, rays are traced in order to create samples in space, or rather in directions from the camera location. For audio these samples are taken in time with no regard to the position or direction of a sample. As a result, and unlike image rendering, the number of samples in the result is unknown at the beginning of the render. Our implementation uses a map to store samples, using an unsigned long as a key, relating to the sample number in time, and storing each sample as a double. The sample number n_s for a time t , at a sampling rate f_s can be calculated with Eqn 5.

$$n_s = \frac{t}{f_s} \quad (5)$$

All of the examples created use a sampling rate of 44.1 kHz.

Once the desired samples have been stored in the map, the program writes the samples to a wav file. The file format consists of a header format indicating information about the file, such as number of channels, sample rate and file length, and then a list of samples, in order. The program writes the header information to the file, and then writes all of the recorded samples, filling in unwritten locations with zeros, until it reaches the last sample with a value greater than zero. The samples are written into the wav format as integer values, so the samples are scaled and converted as they are written.

The implementation for the storing of samples and writing to an output file is written in the WavWriter class.

4.2 The Image Method

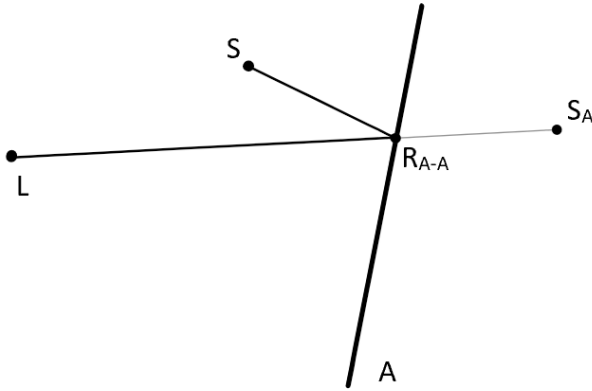
The image method provides a model for calculating perfect specular reflections across an arbitrary surface, from one point to another. The reflection distance and point of intersection with the surface can easily be modeled by creating an image of the source point, reflected over the surface. The line traced from this image point to the destination point has the same length as a reflection across the surface. The point at which this line intersects with the surface is the same point at which the ray would reflect from the original point [4]. A simple reflection with one surface is shown in Fig 3. Reflection of a point S across a surface A is accomplished by creating a ray originating at S in the direction of the normal of surface, n_A . The plane intersection of the ray is calculated. It does not matter if the ray intersects with the surface itself, only the plane on which the surface exists. In addition, the sign of the intersection point does not matter, it can be "behind" the point. The image point S_A is then calculated using the distance of the intersection d in Eqn 6.

$$S_A = S + 2dn_A \quad (6)$$

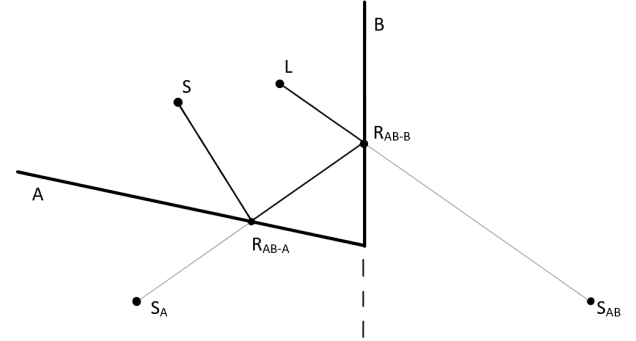
Material	Attenuation Constant (dB/m)						Propagation Velocity (m/s)
	125 Hz	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	
Air	0.000	0.001	0.003	0.005	0.010	0.030	343

Table 1: Travel Media used in simulations

Material	Absorption Constant						Scattering Coefficient
	125 Hz	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	
Wall Absorber	0.09	0.30	0.85	0.63	0.38	0.47	0.4
Ceiling Absorber	0.45	0.45	0.52	0.60	0.50	0.60	0.6
Chair Cushions	0.13	0.45	0.75	0.94	0.94	0.91	0.8
Wood (Chair)	0.09	0.11	0.09	0.07	0.06	0.07	0.8
Concrete	0.01	0.01	0.02	0.02	0.02	0.03	0.2
Plaster	0.10	0.10	0.10	0.10	0.10	0.10	0.4
Carpet (Stairs)	0.03	0.04	0.09	0.20	0.34	0.45	0.7
Carpet (Floor)	0.03	0.04	0.09	0.20	0.34	0.45	0.2
Wood (Table)	0.09	0.11	0.09	0.07	0.06	0.07	0.4
Wood (Doors)	0.09	0.11	0.10	0.07	0.06	0.07	0.2
Drywall	0.19	0.14	0.09	0.06	0.06	0.05	0.2

Table 2: Reflection Materials used in simulations**Figure 3: S is reflected over the surface A to create the image S_A , which can be traced to the point L through reflection point R_{A-A}**

Using this method to calculate perfect specular reflections, we can build an impulse response of a room containing these perfect materials. For each line generated by this method, the line must be tested for intersections with surfaces in the room. In order to be considered a valid intersection, the ray must intersect with the reflecting surface, and must not intersect with any other surfaces in the line segment between the reflection point on the surface, and the destination point [4]. In addition, there cannot be any intersections between the initial point and the surface intersection point. This method extends simply allow for multiple reflections by creating additional images. To model a reflection from S, reflecting off of

**Figure 4: A multiple surface reflection, from point S to surface A to surface B to point L**

surface A then off of surface B, the image S_A is created using Eqn 6. Then the next reflection can be modeled by reflecting S_A across B to create S_{AB} . To calculate the full reflection, first the line is traced from destination L to S_{AB} , calculating the reflection point R_{AB-B} . Then a line is traced from R_{AB-B} to S_A , calculating reflection point R_{AB-A} . Finally, the line is traced from R_{AB-A} to the original source S. Fig 4 shows a multiple surface reflection as described above.

For our implementation we use node based tree structure in order to calculate store the potential bouncing geometry for this image method. This helps to keep track of the path that a bouncing particle will take in order to reach its destination, and to prioritize the calculation for each reflection. Each node contains a pointer to its parent, a pointer to the face across which this node is reflected, the point of the image associated with this reflection, and the depth of the node in the tree. A priority queue is used to determine which

node is processed next, sorting on the depth in the tree. This is done as reflections with fewer bounces tend to more prominent and therefore more important to the final result [6]. When first used, the node tree consists of a single node, with no parent or face, and positioned at the source.

When a node is processed, first it creates a new node for each face in the scene, by reflecting over each face except for the face associated with the current node, if one exists. These new nodes have the current node as their parent. Next the node being processed works its way up the tree, calculating the reflection points for each surface until it reaches the source, which is the root node of the tree structure. At each reflection, it scales the current energy of the sound wave in accordance with the absorption constant of the material associated with the reflection face. The total distance traveled is recorded as tree is traversed. When the full path of bounces has been calculated, the energy is scaled by the attenuation constant of the travel medium by the distance traveled. The remaining energy is added to the response at the time which can be calculated using the distance and the propagation velocity from the travel medium.

In the current implementation the program is stopped and the output file is written after a certain reflection depth is reached, determined by a command line argument. The implementation of the Image Method is written in the `Imager` class.

4.3 Pathtracing

Pathtracing provides a much more flexible model for calculating reflections, that allows for the simulation of both specular and diffuse reflections. Pathtracing is a Monte Carlo approach that simulates an impulse by generating a large number of random rays from the source point, and follows the path of each of these rays and records their interactions with the listening point. Since we are modeling random rays, the chance that any would intersect with the listening point is almost nonexistent. In order to remedy this, the listening point is simulated using a sphere as is described by [4]. The usage of a sphere creates some error in the simulation, but careful choice of the size of the sphere allows for the creation of believable results. The size of the sphere is determined as a command line argument in our implementation.

The direction of the rays does not matter, and should be uniform across the unit sphere unless we wish to model a directional sound source. For the purposes of this paper we assume that all sound sources have no directional element.

For each ray that is generated, the path is traced as it travels, determining the first intersection point along its path with the scene geometry, and determining if the ray intersects with the listening sphere. If the path does intersect with the listening sphere, the distance is compared to the first intersection point with scene geometry, to ensure that the sphere is encountered first. If this test passes, the ray adds energy to the response. The time delay of the system is calculated the same as in the image method. The energy transferred for a given frequency f is calculated using Eqn 7 where d is the distance from the start of the ray to the intersection point with the sphere, E_r is the starting energy of the ray, n_r is the ray direction, n_s is the direction from the center of the sphere to the intersection point, k is the number of samples, and α_f is the

attenuation constant for the travel medium at f from Eqn 3.

$$E_f = \frac{E_r 10^{d\alpha_f/20} |n_r \cdot n_s|}{k} \quad (7)$$

The dot product of the two normals is used to scale the energy transfer by how much of the sphere is actually intersected by the ray, the same way as described by [4] but with different methodology.

The intersection point with scene geometry is used to generate a reflection point and create another ray. Unlike raytracing for the rendering of images, a ray will always bounce and generate another ray until a defined capping point. For our implementation we use a maximum bounce depth defined by a command line argument, and also stop bouncing if the energy of the ray has decreased by more than 60 dB from the original impulse. There are two types of reflections that can take place in this simulation, specular and diffuse. A specular reflection is for a surface with normal n_s calculated using Eqn 8.

$$n'_r = n_r - 2n_s(n_r \cdot n_s) \quad (8)$$

A diffuse reflection is modeled by creating a new weighted random direction, calculated with Eqn 9, using a random normal vector n_{rand} .

$$n'_r = \frac{n_s + n_{rand}}{\|n_s + n_{rand}\|} \quad (9)$$

The scattering constant of the intersected surface material is used to determine if a reflection will be specular or diffuse. Unlike raytracing methods, each intersection results in a single outgoing ray when using the pathtracing method. Each bounce uses the new direction and starting point the recursively call the ray pathtracing function this time with a starting energy E'_n calculated by Eqn 10 from the previous ray starting energy E_n , Eqn 3, and Eqn 4.

$$E'_r = \beta E_r 10^{d\alpha/20} \quad (10)$$

The implementation of pathtracing is written in the `PathTracer` class.

4.4 Rendering Audio Using Impulse Responses

As described above the impulse response is a representation of the behavioral response of a system, and can be used to calculate the response of any input signal in that system. This output response is calculated by convolving the input signal by the impulse response $H[t]$ of the signal $S[T]$ using Eqn 2.

The rendering of audio is performed in real time using a program written in Pure Data. The program loads in six impulse responses from wav files into array buffers which can be used later in the program. Convolution is performed using the `convolve~` object from pd-extended using a 256-bit buffer. This buffer can be made smaller or large, where a smaller buffer results in less accurate results with lower latency. The 256-bit size was chosen to be the best balance between latency and performance. Each frequency has a separate impulse response, and therefore each frequency requires a separate `convolve~` object. The input signal is separated into six frequency bands for each separate impulse response. This is accomplished by using a 2-pole lowpass filter with both poles located at 187.5 Hz for the 125 Hz response, and a 2-pole highpass filter with both poles at 3000 Hz for the 4000 Hz response. The internal frequencies are separated using 4-pole bandpass filters with two poles each at the upper and lower bounds, separating the

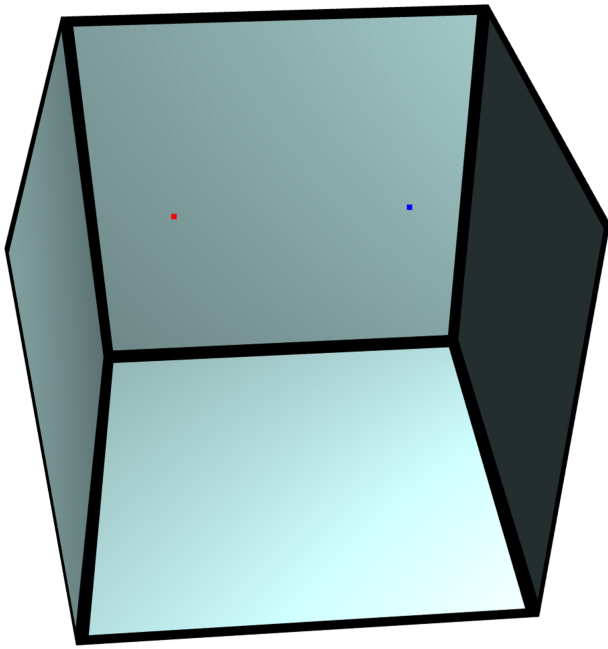


Figure 5: Two meter cube with source point in red and listening point in blue

signal into bands of 187.5 Hz to 375 Hz for the 250 Hz response, 275 Hz to 750 Hz for the 500 Hz response, 750 Hz to 1500 Hz for the 1000 Hz response, and 1500 Hz to 3000 Hz for the 2000 Hz response. The resulting signals from each convolution are added together to create the output signal. The resulting output was compared with the input using an impulse as the impulse response, where the expected output should be the same as the input. The result was extremely similar and proved difficult to tell the difference for most observers. The latency of the system was fairly low and proved suitable for live performance.

5 RESULTS

We have run the program on a number of different scenes and compared the results to one another, by reviewing a visualization of the responses, and listening to impulse responses, and results of convolution of sound with the results. The image method did not produce results capable of producing reasonable output sounds for many of the cases, though they provided a good reference for reflection times.

5.1 Small Cube

The first example uses a small cube for the mode. The cube is two meters in all three dimensions, and uses the material properties for Wallpaper from Table 2. The source and listener are placed close to the center of the cube, one meter apart, and 1.5 meters above the ground. A visual of these locations in the scene geometry is shown in Fig 5. The simulation for this was run using a listener sphere with a 0.1 m radius, a maximum reflection depth of 100 bounces, and using 50,000 sample rays.

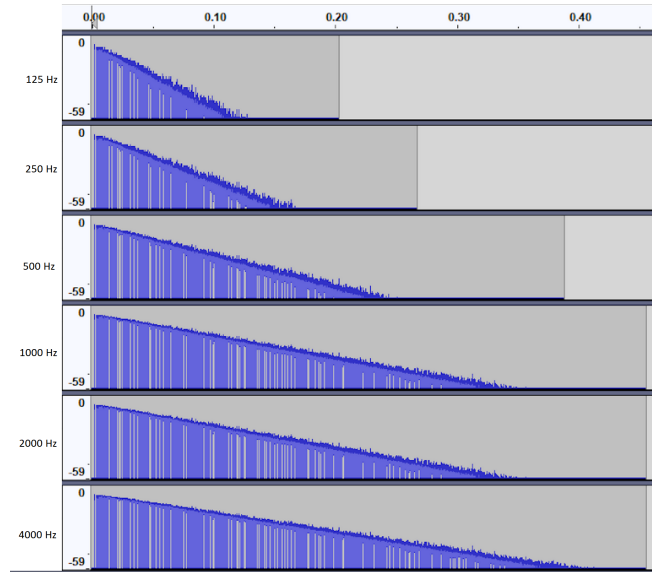


Figure 6: The impulse response for the two meter cube at six frequencies

The response for this example is shown in Fig 6. The 60 dB reverberation time for this example is 0.451 seconds. When used with the sound rendering program, this simulation presents reasonable results. There are no extended echoes and the closeness of reflections creates an effect that is easily replicated by playing sounds in a small reflective closed space. In addition, the small space of the environment means that most of the lost energy is due to absorption by reflection materials rather than attenuation by travel media, which results in a more prominent concentration on upper frequencies.

5.2 Medium Cube

The next example uses a larger cube, this time twenty meters in all dimensions. The materials for all faces are unchanged, using the Wallpaper material from Table 2. The source and listener are placed close to the center of the cube, one meter apart, and 1.5 meters above the ground. A visual of these locations in the scene geometry is shown in Fig 7. The simulation for this was run using a listener sphere with a 0.5 m radius, a maximum reflection depth of 100 bounces, and using 100,000 sample rays.

The response for this example is shown in Fig 8. The 60 dB reverberation time for the example is 2.146 seconds. This result creates a very clear echo, as opposed to the previous example. The differences in times for the upper and lower frequencies is not as extreme as in the smaller cube, as the upper frequencies are attenuated more drastically with the longer travel distances.

5.3 Large Cube

This example shows an extreme example for simple geometry. The geometry is a much larger cube, one hundred meters in all dimensions. The materials for all faces are unchanged, using the Wallpaper material from Table 2.

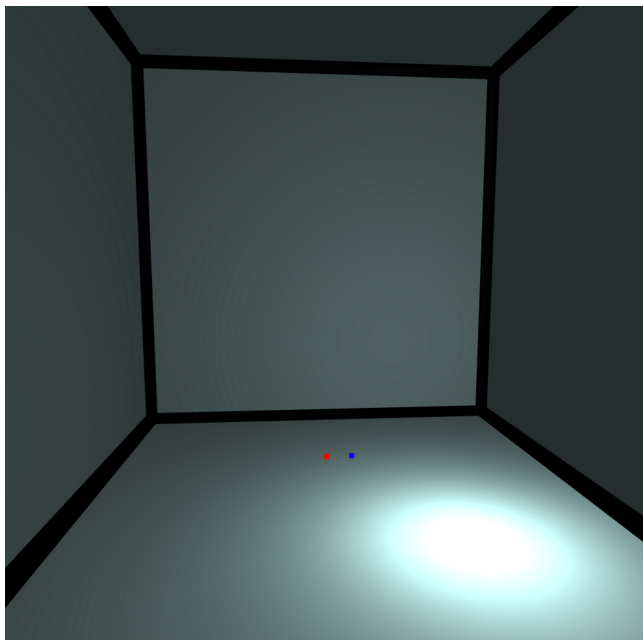


Figure 7: Twenty meter cube with source point in red and listening point in blue

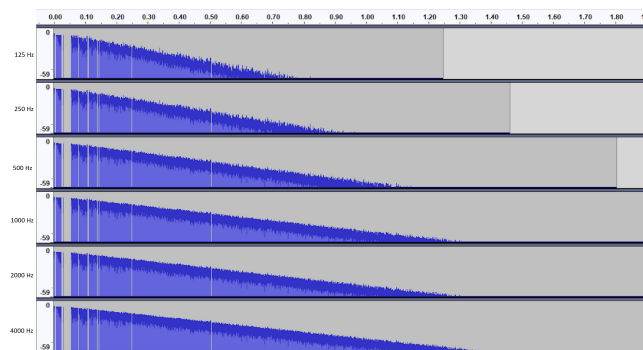


Figure 8: The impulse response for the twenty meter cube at six frequencies

The source and listener are placed close to the center of the cube, forty-one meters apart, and 1.5 meters above the ground. A visual of these locations in the scene geometry is shown in Fig 7. The simulation for this was run using a listener sphere with a 2 m radius, a maximum reflection depth of 100 bounces, and using 1,000,000 sample rays.

The response for this is much more extreme than the previous response, and behaves as expected. The 60 dB reverberation time is 2.852 seconds, and the differences in reverberation times between frequencies are almost the same. In the impulse response in Fig 10, there is a clear spike at 0.12 seconds representing the delay before the direct transmission of sound as opposed to the much smaller delay of about 0.002 seconds in previous examples. This initial delay is accurate as the accurate response time can be calculated

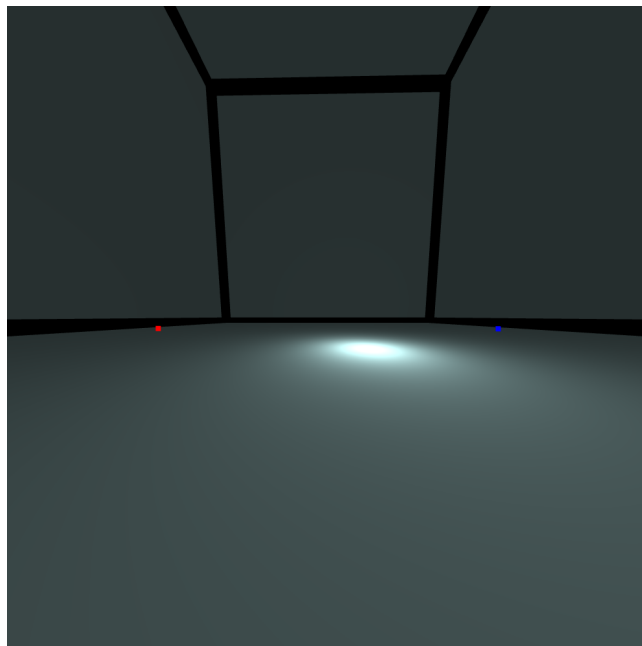


Figure 9: One hundred meter cube with source point in red and listening point in blue

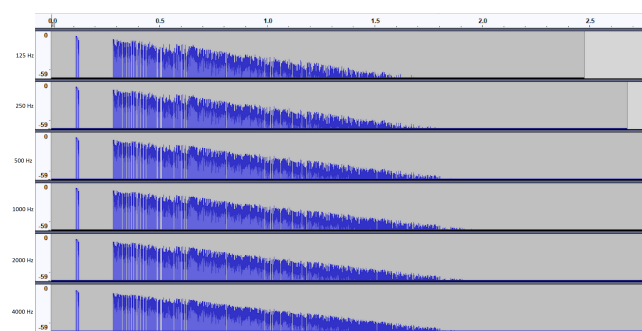


Figure 10: The impulse response for the one hundred meter cube at six frequencies

using the distance, 41 m, and propagation velocity, 343 m/s, to calculate the perfect transmission time to be 0.11953 seconds. The choice of listener sphere size and the number of sample rays was a key factor in acquiring a realistic response. The response for this example works extremely well for most input sounds, though some extremely short sounds will result in a sort of pixelation of the output, representing spaces in time where there was no ray that hit the listener, though there would be sound heard then in an accurate portrayal.

5.4 Room With Varying Materials

The purpose of the next example is to show the importance of reflection material choice and how they effect the output of the simulation. The models for these tests are geometrically identical

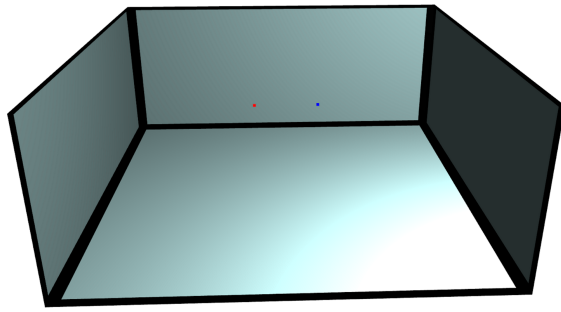


Figure 11: Geometry of the room

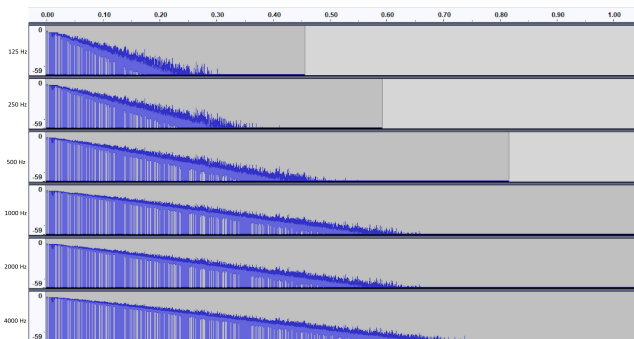


Figure 12: Impulse response for the basic room

and have a square floor plan six meters by six meters, and a 2.5 meter high ceiling.

For all of the simulations in this example, the source and listener are located in the same locations, one meter apart in the center of the room, 1.5 meters above the ground. Each simulation is run using a listener sphere with a 0.5 meter radius, and 100,000 sample rays.

The first simulation is run using the Wallpaper material from Table 2 for all of the faces in the room. The response is shown in Fig 12.

The result is quite similar to responses from the other examples. The 60dB reverberation time is 1.149 seconds, and the longer lasting reverberations are focused in the the higher frequencies. The Wallpaper material mostly absorbs lower frequencies, and leaves the higher frequencies alone, causing the higher frequencies to last longer in a space of that size. The next simulation replaces the floor material with the Carpet (floor) material from Table 2. The response is shown in Fig 13.

This seemingly small change has a major effect on the response of the room. The 60 dB reverberation time for this simulation is 0.781 seconds, a 32.0% decrease. The carpet material absorbs a lot of upper frequency energy, and absorbs less lower frequency energy. This causes a slightly stronger lower frequency response, and greatly shortens the upper frequency response. The regular triangular shape of the upper responses are now gone, and there is great irregularity of the strength in the late reflections. Some are the same strength as before, as they only bounced off of the walls

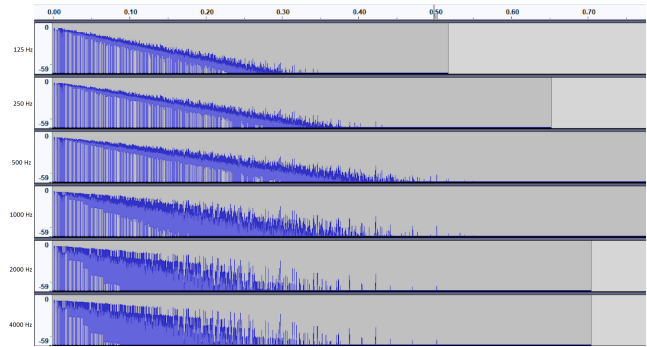


Figure 13: Impulse response from the room with a carpet on the floor

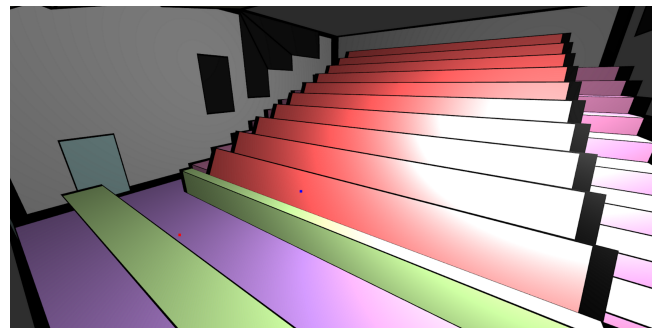


Figure 14: View from the front of the auditorium, with the listener in the front of the audience

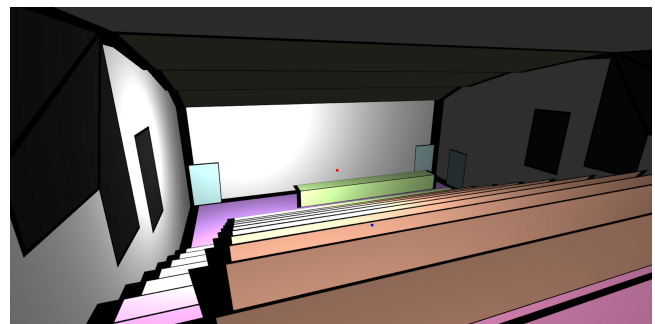


Figure 15: View from the back of the auditorium, with the listener in the rear of the audience

and ceiling, but a lot have bounced off of the floor and have greatly decreased energy.

5.5 Auditorium

The final example puts together a number of concepts from the other examples, looks into the importance of positioning in a simulation. The scene consists of a low-polygon auditorium consisting of ten rows of raised seating, desks, and a shaped ceiling. Fig 14 shows the front view of the geometry of the auditorium, colored by material, and Fig 15 shows the back view.

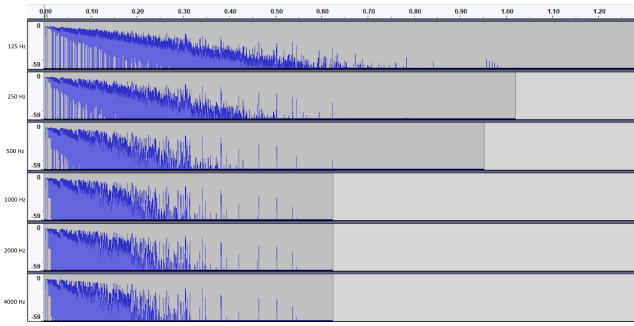


Figure 16: Impulse response of the auditorium, with the listener positioned in the front of the audience

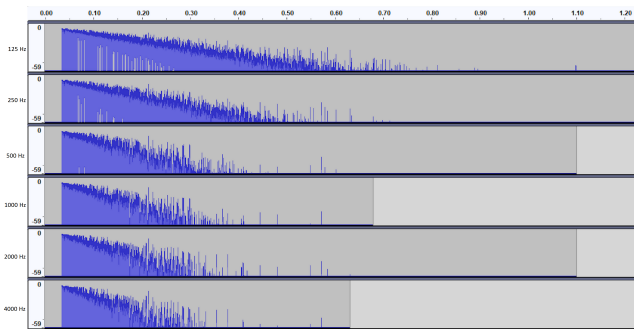


Figure 17: Impulse response of the auditorium, with the listener positioned in the back of the audience

The scene contains a large variety of the materials from Table 2. The floor uses the Carpet (Floor), stairs use Carpet (Stairs), the curved portion of the ceiling uses Ceiling Absorber, the walls use Concrete and the ceiling uses Plaster. The front of the chairs use the Chair Cushions, the back of the chairs uses Wood (Chair), doors use Wood (Doors) and desks use Wood (Table). Absorbers on the wall use Wall Absorber. All simulations for this example use a listener sphere with a two meter radius, a maximum reflection depth of 100, and 500,000 sample rays.

All simulations for this example position the source at the front of the room, at the front desk. The listener for the first simulation is positioned in the center of the front row of the audience. The simulation results can be seen in Fig 16.

The response has a 60 dB reverberation time of 1.304 seconds, which is reasonable when listened to. The variety of materials creates a very interesting response shape, and the longer responses are focused in the lower frequencies, though there are a few later responses in the higher frequencies, most likely originating from reflections off of the walls and non-absorptive parts of the ceiling.

The next simulation positions the listener at the right of the back row of the audience, as is shown in Fig 15. The simulation results are shown in Fig 17.

This simulation has a 60 dB reverberation time of 1.341 seconds, very similar to the first simulation. The most prominent difference that can be seen is the increase in the time delay before sound starts to be heard. Different reflections become more prominent in this

simulation, primarily those reflecting off of the chairs are much stronger, as the sound waves are reflecting off of the reflective backs of the chairs, rather than the absorptive fronts of the chairs like in the first simulation. The responses are much smoother and more regularly shaped in this simulation, which results in a much clearer output when a sound is run through the sound rendering program.

6 LIMITATIONS AND ISSUES

The main limitations of the program derive from the assumptions made for each of the implementations, and many can be remedied by further work.

One major issue with both of the impulse response rendering methods presented in the project are that they rely on the assumption that the geometry is much larger than the wave. This is true for the higher frequencies, but for lower frequencies the waves are much larger and therefore comparable in size with the geometry of the environment [4]. When the size of waves becomes similar to the size of geometry diffraction starts to occur, a property where a wave bends around a sharp edge when only part of the wavefront hits the geometry.

There are a number of potential solutions to this issue. While fluid simulations are too complex and unreasonable for full spectrum simulations, as the simulation is limited to lower frequencies, resolution required for fluid simulations decreases, making them a possible choice for simulating lower frequencies more accurately, which could be combined with the more accurate upper frequency simulation using this technique.

Another potential solution to this issue would be to use a similar technique as in [2] to generate surfaces for sharp silhouette edges around geometry, relative to the starting point of a ray. These faces would be generated to extend both into the appropriate face and away from the edge, so that an intersection would occur when part of the wavefront but not all of it would hit the face, the cause of diffraction. This method would introduce a significant increase in time complexity if silhouette edges were recalculated on every bounce, but there are potentially ways to decrease this by pre-computation of several basic perspectives.

6.1 Issues With the Image Method

The image method works well for the calculation of early reflections with high accuracy, however it presents several limitations for most other cases. The image method is only able to capture specular reflections, and a significant portion of effects desired by this project involve diffuse reflections. None of the materials used or found in related works had a scattering constant of zero, which would be an ideal material to simulate with this method. In addition, the image method does not scale well with the number of reflections. For each additional reflection to be calculated, the image method running time multiplies by the number of surfaces in the environment. The exponential running time makes it unreasonable to render scenes with more than eight bounces, even with geometry as simple as a cube.

The results with the image method tend to have spaced out reflections, such as in Fig 18 using the model from Example 2 with a bounce depth of seven. Even that few bounces results in 117,187 rays being cast, with poor results compared to a similar number

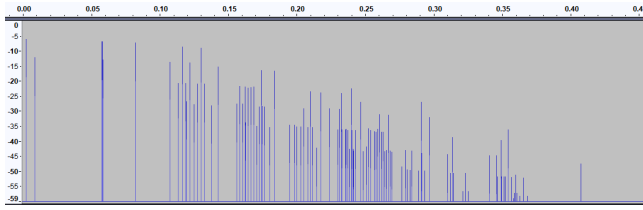


Figure 18: Single frequency response with the Image Method with a bounce depth of seven

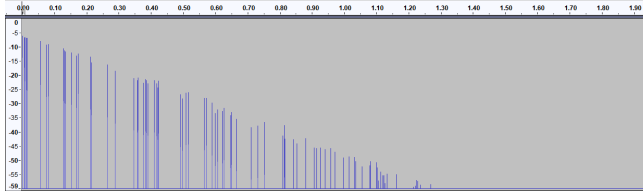


Figure 19: Single frequency response for the 20m cube geometry using a listener sphere with a radius of 0.1 meters

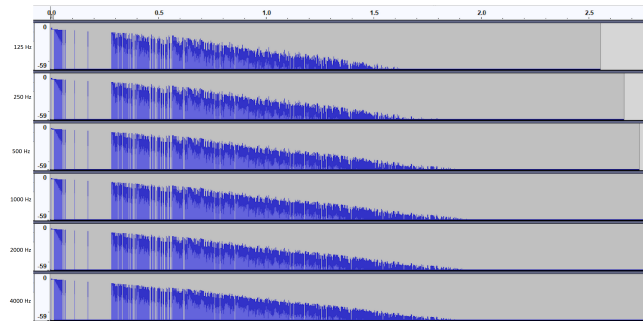


Figure 20: Response for the one hundred meter cube when the source and listener are one meter apart

cast for the pathtracing implementation. This makes it unsuitable for rendering audio output as it tends to create pixelated sounds.

6.2 Issues with Pathtracing

The main issue from pathtracing is the uncertainty and randomness that can sometimes cause poor results. This can be somewhat remedied by adjusting the number of samples and the size of the listener sphere. In Fig 19, a single frequency response is shown for an example where the listener sphere is too small.

Without a sphere of ample size, the response has very sparse sound responses, which is not at all consistent with the real response of the system. This can be remedied by increasing the number of samples, at the cost of render time, or by increasing the size of the listener sphere. There is additional error introduced when the listener sphere is too big however. Fig 20 shows the response from Example 3 when the source and listener are one meter apart.

There are several issues with this example. The first is that due to the size required for the listener sphere, the source is actually located inside the sphere, and so all sound rays create an initial

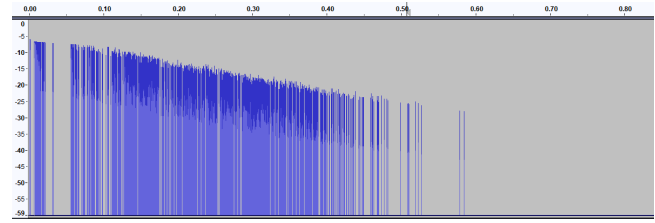


Figure 21: A single frequency response from the geometry for the 20 m cube using a maximum bounce depth of ten

direct response. There are so many rays creating this response that they create a long flat bar for the initial direct response, when it should be much closer to an impulse. This shaping is even seen in other examples where the source is not inside the listener, but the sphere is still too big.

A potential solution for this issue would be for a dynamic sizing of the listener sphere, based off of the distance between the starting point of the current ray and the listening point. This would provide a balance where rays in larger geometry would still have a chance to hit without creating the distortion from rays originating close to the listener.

The other issue with this example is that there are two high amplitude spikes between the direct response and the delayed echo, at about 0.11 seconds and 0.17 seconds. These are not an ideal result but rather they come from diffuse reflections that would normally have a low energy transfer. This is a natural result of the Monte Carlo method that allows for occasional low probability results to come through.

A possible resolution to this error would be to implement a method more similar to traditional raytracing, where diffuse and specular rays are generated, with energy scaled to the probabilities. The issue with this resolution is that this quickly results in a large number of additional rays to be generated, as this method requires a large number of bounces for accuracy. A sample error output where the maximum bounce depth is too low is shown in Fig 21.

7 CONCLUSION

We have presented a program that uses traditional raytracing techniques for rendering impulse responses for simulated environments. Our method creates reasonable results for the purposes of rendering output audio for recording or performances, but tends to stray from accuracy in lower frequencies. In addition, this method has the same pitfalls as traditional raytracing; it scales linearly with the number of surfaces in the scene and can require a large number of rays in order to create a suitable output. The results from this rendering can be used with our other program to render the system response from any input audio in real time.

This program uses traditional raytracing techniques, and therefore can use any number of optimizations for the raytracing methods that have been presented in past years. We would also like to further evaluate the accuracy of the program by comparing the results of a simulation with results obtained by other more traditional techniques for acoustic mapping.

REFERENCES

- [1] Christian Loftin Carl Schissler and Dinesh Manocha. 2017. Acoustic Classification and Optimization for Multi-Modal Rendering of Real-World Scenes. *IEEE Transactions on Visualization and Computer Graphics* 24 (2017), 1246–1259. Issue 3.
- [2] Eric Chan and Frédo Durand. 2003. Rendering Fake Soft Shadows with Smoothies. In *Proceedings of the 14th Eurographics Workshop on Rendering (EGRW '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 208–218. <http://dl.acm.org/citation.cfm?id=882404.882435>
- [3] Steven Cummer, Johan Christensen, and Andrea AlÅž. 2016. Controlling sound with acoustic metamaterials. *Nature Reviews Materials* 1 (02 2016), 16001. <https://doi.org/10.1038/natrevmats.2016.1>
- [4] David Oliva Elorza. 2005. *Room Acoustics Modeling Using the Raytracing Method: Implementation and Evaluation*. Licentiate Thesis. University of Turku. citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.43&rep=rep1&type=pdf
- [5] ISO. 1993-06. *Acoustics – Attenuation of sound during propagation outdoors – Part 1: Calculation of the absorption of sound by the atmosphere*. Standard ISO TS 9613-1:1993. International Organization for Standardization. <https://www.iso.org/standard/17426.html>
- [6] J.H. Rindel. 1995. Computer Simulation Techniques for Acoustical Design of Rooms. *Acoustics Australia* 23 (1995), 81–86.

A ALLOCATION OF TIME

It is difficult to estimate the amount of time spent on this project, but split across project development and debugging, report writing and presentation preparation, about 110 hours total were spent. All work on this project was done by Jacob Dorskocil.