Interactive Visualization of Acoustics

Max Espinoza

Rensselaer Polytechnic Institute
espinm2@rpi.edu

Abstract

Interactive visualization of sound propagation in architectural spaces offers architects an iterative approach to construct rooms with desired acoustical proprieties through changes in geometry and materials in simulated models. We are investigating methods to simulate and visualize sound propagations with interactive frame rates for varying geometries. Simulating the propagation of sound can be done in real time with existing numerically based algorithms, however these algorithms require intensive offline computation for each new room geometry used. This is a geometrically based approach that can simulate sound propagations at interactive frame rates for new geometries. Wave particles [1], introduced by Cem Yuksel for real-time water simulation through rendering extended height fields, can be used to simulate the wave front propagations of sound offering visualizations at interactive frame rates. We created a design tool for architecture students that will allow the creation of 3D models and interactive visualizations of sound wavefronts in created spaces given an source position. With this toolkit designers will be able to assess the general acoustical properties of a room in the early design phase, saving both time and capital in the overall planning of an architectural space.

I. Introduction

Architectural acoustics is a branch of acoustics engineering that relates to the science behind the creation of sound in architectural spaces. Both architects and acoustic consultants are involved with architectural acoustics.

Professional consultants evaluate the acoustics of an architectural space in order to modify sound properties within the space. This is done through use of sound absorbers, changes in floor and ceiling materials, or other sound altering techniques. On the other hand, architects take into consideration the acoustical properties within the early stages of the design and have far greater control over the geometry and location of rooms than those consultants fixing unwanted sound properties post construction. Architects meeting clients needs for noise control, preemptively knowing the location and functions of rooms, can move arrange room locations in order to achieve noise isolation. In addition if architects are planning for sound to travel further into a room they can design room geometries and materials for reverberations.

The advantages of being able to plan and estimate the acoustical properties of a given space before construction are two fold. Firstly, unwanted acoustical properties can be detrimental to building residents and businesses. For example echoes in the workspace or loud street sounds in a residential apartment can causes a significant reduction in the utility of that space. Secondly, many of these mistakes must be later fixed through use of acoustical absorbers, special drywall materials, and insulation. All of which cost the client capital. Hope Bagenal author of Practical Acoustics [2] published in 1940s writes that âĂIJstructural sound insulation has been little understood but remains technically difficult and expensive.âĂİ He goes on to state that âĂIJit is cheaper to conquer noise by segregation and separation.âĂİ While sound insulation is much better understood since the 1940s, it is still cheaper to plan for given acoustical properties of a space then fix sound related problems post construction.

The planning for acoustical properties in a given space is no easy task. Acoustics are hard visualize and understand especially for complex spaces due to sound wave behaviors such as reverberation, diffraction, refraction and absorption. While architects are trained in the basic of sound behavior, the use computer software to run acoustical simulations on their virtually created spaces results in more accurate predictions of acoustical properties.

Simulation software such as Grasshopper [3], a plugin for Rhino 3D modeling tools, offer designers the ability to run acoustic simulations on virtually designed spaces. Given that the architectural design process is iterative, simulations must be provided interactively to promote creativity and allow designers to explore the effects of changing their design on the acoustical properties of their virtually designed space. Rendering the results of these simulations as visualizations or auralization are provide useful feedback to designers. The ability to visualize sound propagation through a given space gives designers insight into improving designs for noise isolations, fixing unwanted reverberations, and planning acoustical reflections.

Architecture 3D modeling software tools such as Rhino, AutoCad Architecture, and AutoDesk Revit Architecture are mature software that contain many features for building models. However due to the complexity of building models using sophisticated software, users tend to take longer periods of time creating simpler models used in the initial exploration of room geometries and layouts. Previous research done for lighting visualizations allows user to use physical primitives to sketch a room model on a tabletop interface. [4] With the extension of a user interface, designers can create closed models through drag and drop 2D representations of those primitives to sketch a 3D geometry of a given room within a few seconds. Furthermore the tools required to build models that most acoustical simulations use as input are time intensive, taking away designers ability to creatively explore renovations on their model at interactive rates.

There exist several algorithms for the simulation of acoustic in architectural spaces, however many of these algorithms geared at interactivity do not simulate some acoustics behavior, such as diffraction. Using models created through placement of primitives in a web interface we explore how to improve existing algorithms such as phonon tracing [5] to handle sound behaviors such as diffraction. We believe that if we can simulate a wavefront through the use of wave particles in a phonon mapping based simulation then we can offer a wider range of visualization while capturing properties that most real-time algorithms for acoustics simulation do not. Since we are focused on the visualization portion of this research for our class final project and wave particles have yet to be fully implemented at the time of writing this report I will focus more on what was discussed during the final presentation.

II. Related Works

I. Phonon Tracing

A visualization technique, partially implemented in this visualization, is phonon tracing [5] introduced by Martin Bertram in 2005. Given a geometry, source, and listener position this simulation method returns a finite impulse response filter. This response filter is the result of sounds complex interaction with the geometry of the room and various surfaces with frequency-dependent absorption coefficients. Furthermore Bertram can use these filters to create an auralizations of sound in the environment. For this project we concern ourselves with the visualization portion of BertramâĂŹs algorithm. Phonon tracing is accurate for midand high-frequency sound, however low frequency sounds are not simulated accurately in phonon tracing. The visualization behind the simulation uses colored blobs to trace the path created by a phonon throughout space as it interacts with a roomâĂŹs geometry. Many of these blobs allow users to see the wavefront propagation temporally. Bertram uses color to visualize the spectral energy of phonons; phonons with low frequency are depicted as blue, medium frequency as green, and high frequency as red blobs. Furthermore the darker the color of a phonon gets represents how much energy has been absorbed by the materials it has been interacted with. A color scheme we base one of our visualization modes on. Changes in frequency happen when phonon interact with materials in the scene that change the reflected phonon spectral energy. Bertram shares our same objective, in that given virtual models we want users, through analysis, to improve the sound properties the space through changing model's geometry and materials. Users can interact with this simulation by using a slider to go through time and visualizing the phonons corresponding to a Dirac [5], a unit impulse, sent out from the initial source.

The biggest limitation of Bertram's method is that it cannot capture phenomena that occur with lower frequency sound such as diffractions. A phenomenon we attempt to capture in our visualization through splitting of phonons as wave particles.

In phonon tracing each photon carries the following information with it: an amount of energy, distance traveled from source, its current position, and outgoing direction. This visualization being an extension of phonon tracing uses the same phonon attributes. However, we take into account splits and merges that occur because of our desire of having more phonon to later create a surface mesh over for visualization purposes, and to better capture sound behaviors like diffraction. Due to this we could not not take shortcuts introduced in this paper to increase the performance of phonon mapping. We use phonon mapping as a base to to implement wave particles and capture more acoustical phenomenon.

II. Wave Particles

Our paper is also an extension of Wave Particles for use in acoustics simulations [1]. Wave particles are originally used to create an extended height field of interactive real-time fluid simulations. Wave particles are implemented to be fast, easy to code, and unconditionally stable way to approximate a wave simulation. Yuksel also accounts for local wave induced flow by warping this extended hight field to better mimic water. With the use of modern graphics hardware to accelerate the process of generating this height field, real time frame rates can be achieved for hundreds of buoyant objects in the water simulation. The cornerstone of this paper is using wave particles to model complex wave interactions by computing the local deviation functions required to solve the wave equation [1]. Usually simulations solving wave equations require more computationally expensive algorithms, however Cem reduces problem by keeping track of wave particles propagating throughout a 2D plane. These wave particles each carry a radii, amplitude, position, and direction. Note that these attributes are similar to those carried in phonon tracing. Cem Yuksel use of wave particles give us the idea of splitting phonon so that we can overcome the under-sampling problem phonon mapping experiences. In addition later we with a uniform phonon map we can wrap a mesh around phonons to visualize the wave front as a smooth surface interacting with the given geometry.

However, Yuksel implements wave particles to be as independent as possible, in order to achieve greater efficiency to render in real time. As a result diffraction is not taken into consideration and ignored. Instead conditions where diffraction might affect the outcome of the simulation are avoided. To handle phenomenon such as diffraction, one must into account each particles neighbors, which is more computationally more expensive. This also reduces the overall efficiency of the algorithm, and making it less parallel and portable to GPU. Considering we are trying to handle diffraction in 3D, this is a major drawback to using wave particles.

III. Implementation

As mentioned previous we wanted to extend wave particles through the use of phonon mapping in a geometry representative of an architectural space. Given that we wanted to incorporate previous research we used models generated by our web based user interface for daylighting simulations.

Firstly, this project required the ability to load a mesh in OBJ standard format and parse it such that we can differentiate between structures in a room, such as the ceiling, individual walls, windows, floor, and extra geometries. OpenGL was used because of existing frameworks and data structures that aided in loading meshes and more importantly OpenGL can handle much larger element on the screen then other graphical tools such as Processing and OpenFrameworks.

Secondly, we needed to implement particles moving within the mesh with collision detection in order to simulate acoustics via phonon mapping. We used the ray tracer data structure provided in previous assignments to aid in this implementation. In addition we needed to incorporate absorptive materials and account for the resulting change in sound intensity.

Lastly, we needed to implement wave particles and how they would fit into phonon mapping. We related individual phonon to the wave particles used in water simulation and showed how despite significant reduction in performance they later can be optimized to account for sound behaviors such as diffraction. The point cloud that wave particles provide can later leverage for other visualization techniques. Such as fitting a surface to create a mesh for the wave front to deliver clearer visuals. Most of this is outside the scope of this visualization class, and will be talked about briefly later in this report.

I. Mesh Visualizations

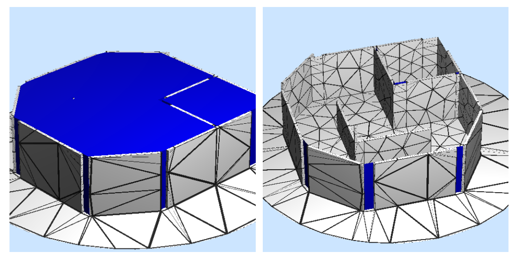

Figure 1: *The wireframe rendering of mesh loaded into OpenGL. Viewing the ceiling is optional.*

We built the application from previous work that could load triangle meshes into OpenGL. Note that we used GLFW and GLM for all my display controls and math libraries, both of which are modern tools for programing in OpenGL 3.0 and newer. One difficulty we encountered was finding a manner to differentiate each wall in the mesh. Since all the walls saved within our OBJ files were categorized as a single wall. Updating an intermediately file to explicitly label each wall with a different material in the OBJ worked. From that we could parse what wall each triangle in the scene belonged too. We created a wire frame visualization to confirm that our mesh is loaded in correctly. In addition we choose the color the geometry in the scene depending on the material it was made of. Currently we only support a small range of materials, however we look forward to supporting more materials in the future and possible overlay of material textures.

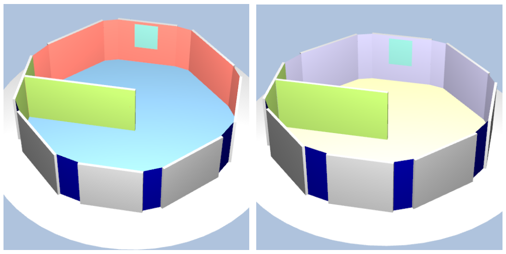

Figure 2: *Left: Room has brick walls & carpeted floors Right: Room has concrete walls & pvc based floors. Both show an absorber wall in the center of the room.*

Table 1: *Material Color Legend*

Material	Color
PVC Plastic Floor Base	Light Yellow
Carpeted Floor	Periwinkle Blue
Brick Wall	Red
Concrete Wall	Violet
Tiled Wall	Gray
Absorber Pad	Light Green
Double Layered Glass	Sky Blue
No Materials	Silver

II. Phonon Mapping

Phonons in Bertram paper carry the same attributes as ours such as: direction, position, intensity, and frequency. For the intensity we use wattage in order to represent an amount of work each phonon is doing on the air around it. Bertram, in aiming to create an audio based aurilization creates his own unit of measurement for phonons, however we choose not to because are currently only interested in visualizing sound waves. Phonons being a representation of work on a medium such as air, are shot into the scene from an initial random sample projected on the surface of a sphere. Depending on the sound source we choose from a range of frequencies to assign each phonon. Phonon's wattage, unit of work, remains constant until a collisions with a material. [1] Then that material adsorbs a specific percentage of a phonon intensity depending on the phonon's frequency. Only the frequency remains static over the simulations as intensity decreases over time.

To save time in computation whenever a phonon is created we calculated how many time steps it would take to reach the closest wall in its path. The phonon then moves each time step until the phonon hit it's closet wall. We then computed the reflection angle and set the phonon for the next collision. Because of this we only get drop in frame rates when a lot of particles hit walls at the same time.

Given we are trying to make a scientific tool, the placement of phonon post collisions are computed from where the it intersected the wall. And then we moved the particle depending on how much time was left after the collision within that time step. This is to ensure accuracy in the simulation. [2]

The Bertram includes piecewise linear functions of material absorption coefficients that we used to compute how the intensity of each particle change as a function of frequency.

Time was also spent exploring different visualization methods, of which proved to be difficult as the range of sound from 20Hz to 20kHz is very large. We then choose to convert the intensity of each particle into a logarithmic scale to give more informative visualizations. Seeing I used wattage to represent the intensity of a phonon I converted wattage to decibels, which relates closely to how we hear sound within the range from 0db to 120dBs.

[1] Note that wattage will decrease with distance from source, however we did not simulate this feature for this assignment.

[2] We are using double-precision floating-point values to store positions and times. And floating-point values for all other attributes in the simulation.

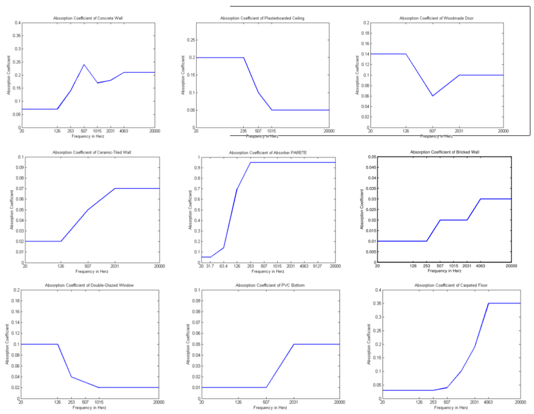

Figure 3: *The absorption functions for the materials we implemented in our simulation. These charts were featured in Bertram phonon tracing paper.[5]*

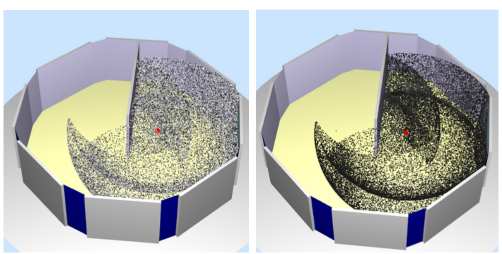

Figure 4: *On the left is the frequency visualization of broadband white noise from 20Hz to 20kHz, on the right is an intensity visualization set at 120 dBs.*

We created four visualization modes of our simulation results. In the frequency visualization mode I mapped white to high frequencies phonons at 20kHz and low frequency phonon to black at 20Hz. This mode is useful for seeing the differences in frequencies distribution between sound sources. As can be seen in Fig 5. Secondly I did a similar mapping for intensity, I mapped black as being the most intense at 120bBs and white as being least intense as 0dbs. I then created a hybrid where I mapped low frequency phonon with to the color blue, and medium frequency to the color green, and higher frequency to the color red. I also wanted to show the intensity of a phonon, which I mapped into the alpha value. Figure 6 illustrates how the hybrid mode is useful in seeing frequency where there are multiple sound sources being tested.

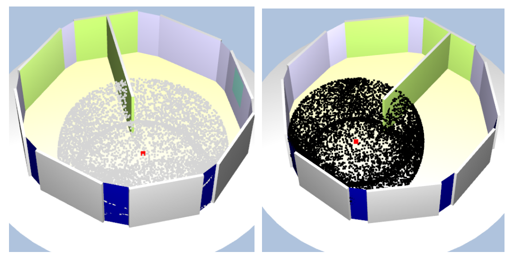

Figure 5: *Left: Visualization of frequency of a low humming AC unit. Right: Visualization of frequency of a high pitched CTR monitor*

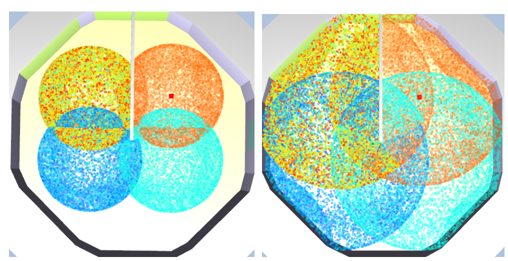

Figure 6: *Left: Visualizing four different sound sources. Right: Continuation of visualization.*

Lastly for the direction visualization I mapped RGB color space to the X Y and Z component of each phonon's direction. This yielded in a smooth transition between directions and makes viewing static images of the phonon map clearer to distinguish between different 3D waves fronts in the simulation.

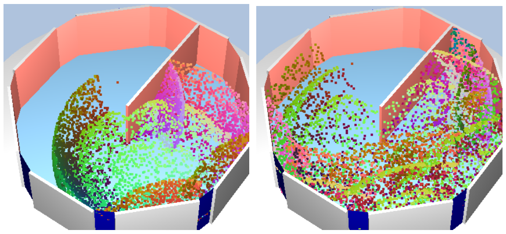

Figure 7: *Wave visualization.*

III. Wave Particles

Wave particles proved to be difficult and more work it needed to fully implement them into our simulation. The wave particles we created caused recursive patterns that are not useful in representing a wave as a collection of points on a sphere. The rules behind when to split a particle, how to split a particle, and how to merge particles were chosen as follows. Firstly, we choose to split a phonon, when that particle was 2 radii away from all other phonons. Moreover, we experimented with other rules to split, none as of yet yield successful results. We also experimented with several split shapes however for future research it is suggested hexagons work best for surface fitting. Lastly, We have yet to implement phonon merges.

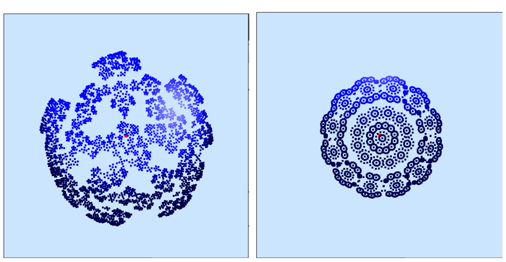

Figure 8: *Recursive patterns from several splits of a single phonon. Note the large empty spaces between recursive patterns. This is not a uniform distribution across a sphere.*

IV. Results & Discussion

Overall for the visualization final project we achieved the goal of setting up the framework for graduate research in acoustical simulations and visualizations. IâĂŹve implemented phonon tracing with material properties on objects in the scene, however I did not finish the implementation of transforming the phonon into wave particles. The splitting of the each wave particle and rules that direct those splits require more work and debugging to figure fully implement. In addition need still implement merging of particles, to prevent the wave fronts from become to dense in their sampling. Despite setbacks we were able to gather some interesting results though, running our simulation for 10,000 iterations we were able to see some expected results in the visualization due to the differences in wall materials and sound source. Below we show some examples and a brief explanation as to why results match our expected institution of the scene.

I. Carpet floor vs PVC floor

Carpets are made up of many fiber filaments that cause diffuse reflection with both light and sound. Carpets across all frequencies also absorbs more sound than plastic flooring. Thus we should see that the intensities of the phonon in the carpeted model on average are lower then in the plastic floor model. This is noticeable in the lighter shade in the intensity visualization in the carpeted model in figure 9. Also noticeable in the less opaque phonon in the hybrid visualization.

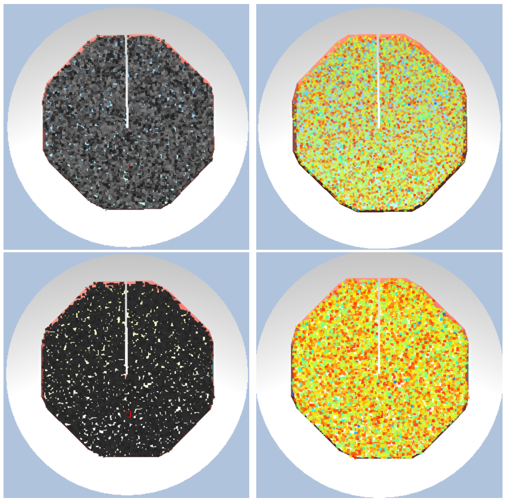

Figure 9: *Top: The intensity and hybrid visualization of a model with carpeted floors Bottom: Same visuals for model with PVC plastics floor.*

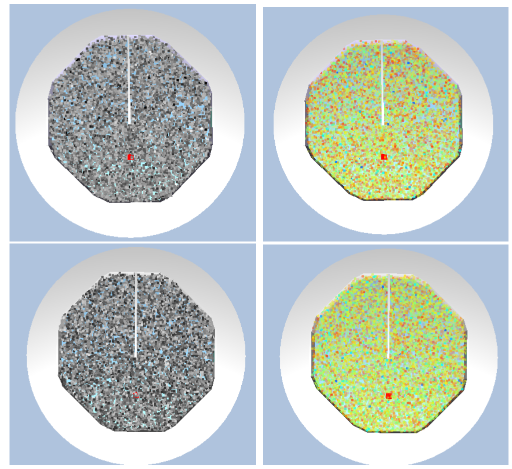

Figure 10: *On the top are the results of a room with cement walls, and on the bottom are the results with tiled walls.*

II. Concrete vs Ceramic Tile

Just as above, according the absorption function ceramic tile is a horrible absorber of sound. Out of all the material tests ceramics are absorb the least amount of noise. On the other hand concrete usually absorbs about a fourth of all sound in most frequencies. So we should expect to see that in intensity of the particles in the concrete figures reflect this. I was surprised to see the results that what looks like similar results in from using both materials. Upon analysis it could be that white noise tends to basis much higher frequencies.

III. Concrete room with absorbers & without absorber

The absorbers are an acoustically engineered material that absorbs about 90% of all sound in wavelengths above 120Hz. In the model with absorbers we placed three in one side of the room to compare with the same model without absorbers. Note both rooms ave concrete walls and share the same floor material. We see that in the room with these absorbers that sound dissipates rather quickly in the radical change between the intensities in both visualization modes.

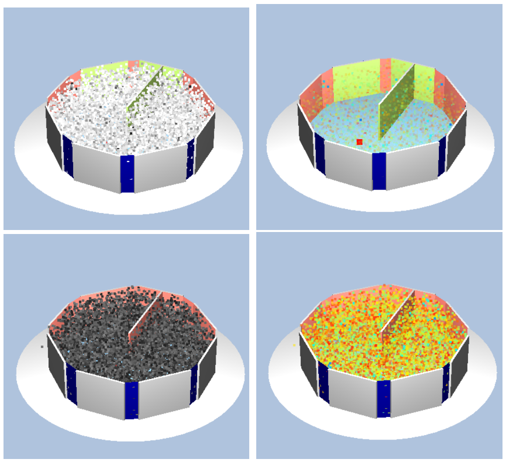

Figure 11: *On the top are the results of a room with absorbers installed, and on the bottom with no absorbers installed.*

IV. Wavefront

Given user feedback, users stated that seeing the direction of the wave front through color would be useful in differentiation when a wave hit a wall and changed direction. The result did make it much easier for me to see wavefront emerge as discrete entities that shared similar color.

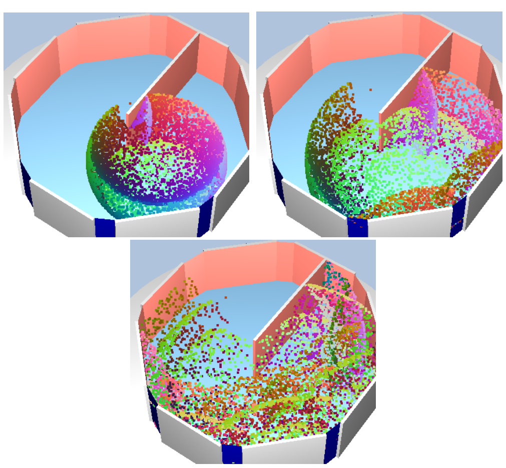

Figure 12: *Wave visualization. Note how you can see the differences in waves color after they hit their surrounding geometry.*

V. Conclusions

Overall, aside from fully implementing wave particles in 3D, I implemented phonon mapping as a base for our wave particle simulation. While I would have like to spent time on using an accelerated data structure to speed up the visualizations to interactive frame rates for over, there is enough content to analysis scene and see see how phonon interact and change at different frequencies depending on the wall material. Future work lies in the completion of integrating wave particles within the simulation in addition to the creating a graphical user interface to control the source sound generated and navigate temporally through the architecture space. Also the addition of a surface mesh wrapped around the wave fronts to better visualize interactions with the scene.

References

[1] C. Yuksel, "Wave Particles, Yuksel Diseration," no. August, 2010.

[2] H. Bagenal, *Practical Acoustics and Planning against Noise*. Brooklyn, N. Y: Chemical Publishing Company Inc, first ed., 1942.

[3] G. A. modeling for Rhino, 2014.

[4] J. Nasman and B. Cutler, "Evaluation of user interaction with daylighting simulation in a tangible user interface," *Automation in Construction*, vol. 36, no. 0, pp. 117 – 127, 2013.

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