Simulating the Reverberation of Cities

Senior Capstone Project Report

Michael Svolos
Stanford University
Department of Music
Center for Computer Research in Music and Acoustics (CCRMA)

Advisor:
Jonathan Abel, Consulting Professor, CCRMA

Abstract: This paper discusses and implements the creation of impulse responses modeled from the geometry of city streets. Impulse responses are created using the image source method, which uses reflection to simulate sound waves bouncing off of flat walls. Using a simplified geometric model of a city street and buildings, simulation of the space is done using MATLAB. Future plans for this model include creating an AU or VST reverberator plugin for more convenient use.

1. Introduction

A few years ago, I was walking in the hills in San Francisco, and I heard the sound of a foghorn from the bay. It was incredible - the horn was so loud, and the way the sound reverberated through the city was unlike anything I'd ever heard. This experience is what inspired me to do more research on reverberation, and is why I decided to pursue this project modeling the sound of reverberation in cities.

At its onset, my goal for this project was to create a reverb AU or VST plugin. After consultation with my advisor about the scope of the capstone, I decided to do the implementation of the project in MATLAB, but with consideration of how the model would look and sound in plugin form, *e.g.* consideration of parameters and user interaction. Nonetheless, the impulse responses created by the MATLAB scripts are compatible with most convolution reverb plugins, so the project still results in sounds that can be used without much difficulty.

This project has to do with science - the physics of sound in spaces - but also with art. In 2020, when anyone with a computer or even a smartphone can make extremely meaningful and beautiful music, the tools and computer programs used for that music become instruments that must be designed as such. As such, I felt that it was important to craft the project with the hope of making it as beautiful and expressive as possible, while remaining faithful to the acoustic models it follows.

2. Background

Reverberation is the quality of a sound that places it in a physical space - what makes a place feel "live" or "dead" (Asselineau, 2015). Its fundamental role in music and sound consumption results in its consideration in the design of nearly all physical spaces, from concert halls to conference rooms. The reverberated element of a sound, distinct from that element of the sound that comes from the sound source itself, is a result of sound waves bouncing off of walls, ceilings, or floors of a room, taking longer paths to reach the listener than the direct path. As sound bounces off of different surfaces, its sound level decreases, and the sound is also filtered based on the qualities of the material, which results in the reverberation "tail", the trailing off of sound after the sound source.

The simplest reverberators use a series of delay lines and filters to represent these multiple sound paths (Smith, 2005). To represent the geometry of a certain space, a geometry-based method of calculating these paths is used, called the *image source method* (ISM). In the image source method, one imagines a sound source, a listener, and the geometry that surrounds them. Sound can be thought of as acting like a ray which obeys the law of reflection when it bounces off of a flat surface. Because of this, one can imagine the listener's position reflected along the wall, and the sound taking a straight path to that virtual listener, rather than a reflected path to the real listener (Mehta, 2015). Using these distances and knowledge of the speed of sound, complex reflections can be calculated and impulse responses for rooms can be obtained.

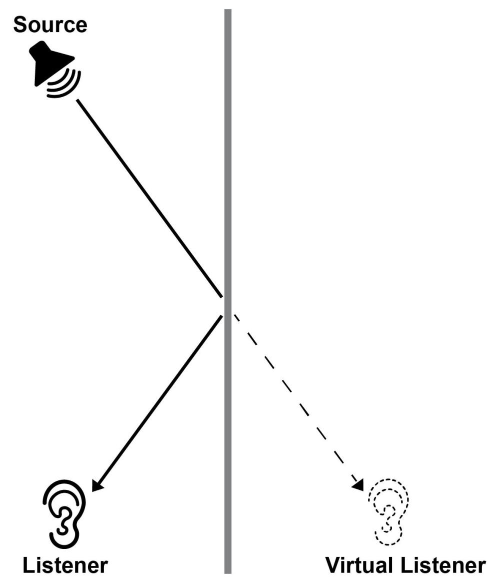

Figure 1: Diagram of source, listener, and virtual listener.

Suppose an impulse is sent from the sound source at time $t = 0$. It travels a distance d from its source to a virtual listener. We then define an intermediate variable $u(t) = t - d/c$, where c is the speed of sound. The dimensions of u are time. We now know that at the sample t for which $u(t) = 0$, there is an impulse that traveled the length of the path from source to listener, and the value of the impulse response at sample t is 1 (before considerations of filtering and gain reduction) (*ibid.*). However, for geometries such as city streets, where building faces are broken up by streets and alleyways, additional checks must be made to ensure that there is a wall to reflect the sound ray at each intersection, which will be discussed later.

3. Methods

3.1 Parameters and city layout

The first major decisions to be made in constructing this project were how the city was to be laid out. Ideally, this layout would be built from parameters that the plugin user could modify to change the sound of the reverberation. To simplify this interaction and the calculations needed, two parameters for the city geometry were chosen - building width and street width. The city then is imagined as a uniform grid of square "city blocks" with the given width, each surrounded by a street of the given width. The sound source and receiver are placed at the center of "crosswalks" at different intersections of the same street. A third parameter, "distance", determines how many blocks away the source and listener are. In plugin form, these parameters would lie within the following ranges:

- Building width: 700 - 1000 feet
- Street width: 15 - 30 feet
- Distance: 0 to 7 blocks. (At a distance of 0 blocks, the source and listener are on opposite ends of the same building, with no intersections between them.)

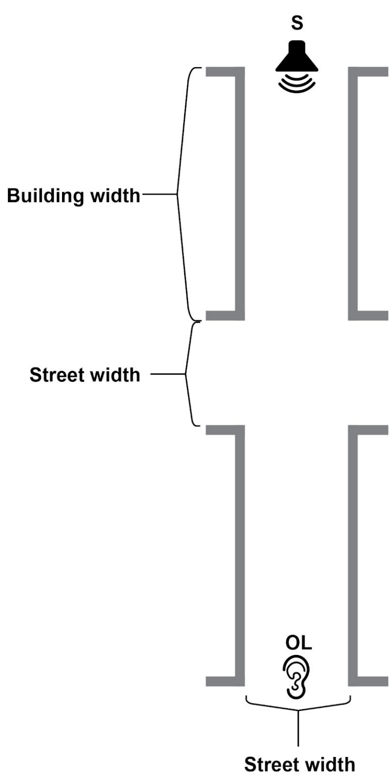

Figure 2: City streets and their parameters, source, and original listener. In this diagram, the parameter "distance" is set to 1; if it was set to zero, the source would shifted downwards by $street\ width\ +\ building\ width$.

3.2 Creating virtual listeners and checking for valid paths

The next step is to construct the sound source, original listener, and virtual listeners for ISM calculations. To start, the source and original listener are placed in the center of the street, with the listener directly "below" the source. The next step is to reflect the listener across any walls that face it. In this case, the only relevant walls are the two vertical walls closest to the listener. Thus, virtual listeners exist evenly-spaced on a line perpendicular to the line connecting the source and the listener, with the width of the city streets between them. In code, these exist as an array of distances from the sound source to each virtual listener, and since the right and left sides are mirror images of each other, only one side is considered. (Stereo impulse responses in this model are created by generating two mono IRs and mixing them according to a given stereo width parameter; due to randomness in the algorithm, these IRs are slightly different and are perceived as stereo.) See Figure 3 below for a visualization of the virtual listeners.

Paths from the source to each VL are checked for time-based and geometry-based validity using the following pseudocode:

```
Initialize empty array “ir”
Initialize array “dists” of distances from S to VLs
For each sample i in ir:
     //check for intersection at this sample
     [value, index] = min(|i/SR - dists/speed of sound|)(1)
     If value < threshold and valid path exists:       (2)
          Filter impulse based on number of bounces     (3)
          Add impulse to ir at i                        (4)
```

Figure 3: Pseudocode for IR generation

Line (1) is the time-based check discussed earlier. `value` represents the function $u(t)$. The value `threshold` in line (2) is initialized at 0.001, since `value` will likely never equal exactly zero. This ensures that the VLs with valid time-based paths will be accounted for.

The rest of line (2), `valid path exists`, makes sure the ray that connects the source and the given VL actually makes bounces with walls every time it crosses out of the street and doesn't go into an alleyway where it wouldn't make contact with a VL. This calculation is done by knowing that the slope of the ray is $(distance\ between\ source\ and\ original\ listener)/(distance\ between\ original\ listener\ and\ VL)$.

Then, the height of the sound ray is calculated for each virtual "wall" that exists between the original listener and the VL, and these heights are taken *modulo* the sum of street width and building width. If any of the results are greater than building width, then that "bounce" would go into a side street and not reflect back towards the listener, and the path is not valid. If not, then the time and location of a valid path is known, and an impulse can be added there. This check is demonstrated in Figure 3.

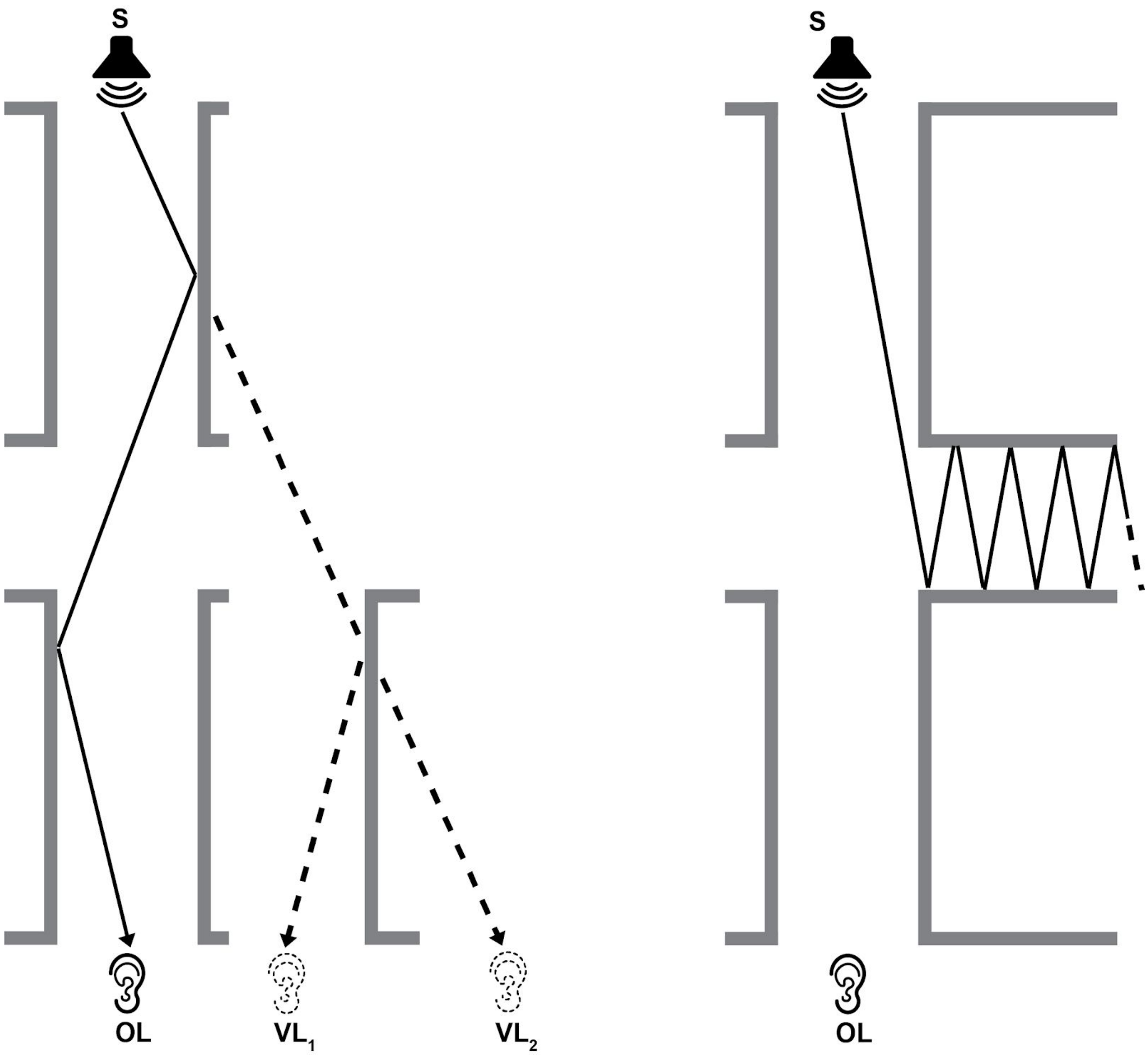

Figure 4: An example of a valid (left) and invalid (right) path from a source to a virtual listener. On the left, all of the sound ray's bounces hit walls, whereas on the right, the bounce goes into an alleyway and cannot reach any listener, virtual or original.

3.3 Filtering impulses

After determining when impulses arrive, these impulses must be attenuated and filtered based on the properties of the materials they bounce off of and pass through. When traveling through air, sound levels decrease by 6 decibels per doubling of distance (*Engineering Acoustics*). Digitally, this can be implemented by scaling the amplitude of the impulse from each ray based on the ratio of the distance between the source and original listener and the length of the given ray.

Impulses are also filtered based on the absorption coefficients per frequency band of common materials used for the outside of buildings. Each impulse is filtered once per bounce it makes to reach its VL. These filters were created using data from JCW Acoustic Supplies and Robert Bristow-Johnson's EQ-to-biquad formulas. For each bounce, the filter to be used was randomly picked from a selection of parameters based on the reflective properties of glass. Since sound is lowpassed slightly as it passes through air, per *Engineering Acoustics*, there was also a chance for a low shelf filter to be randomly selected. Other materials, such as concrete, were auditioned (and are still present in the source code), but were not included in the final version since they resulted in a less desirable sound. Absorption coefficients per frequency for the materials used are shown below.

	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz
Glass	0.18	0.06	0.04	0.03	0.02	0.02
Glass - variant 2	0.09	0.03	0.02	0.015	0.01	0.1
Low shelf	0.4	0.1	0.05	0.1	0.25	0.5

Figure 5: Absorption coefficients.

3.4 Randomness

There are several places in the model where random elements are introduced. These include the following: small random +/- in frequency and gain for each peaking filter and shelf, recalculated each time they're used; random choice of filters for each impulse bounce; and after each impulse is filtered and placed, additional randomly-timed copies of the impulse placed within a few thousand samples of the original impulse. Each of these choices was made in an effort to make the impulses seem more organic - different buildings have different absorptive properties, which is accounted for in small changes to each filter's coefficients. This also serves to avoid accumulation of resonance from repeated instances of the same filter. As an example, shown below are frequency responses form 25 randomly-generated filters based on the "glass" coefficients. Note that though there is variation in the exact values of the responses, their general shape is the same.

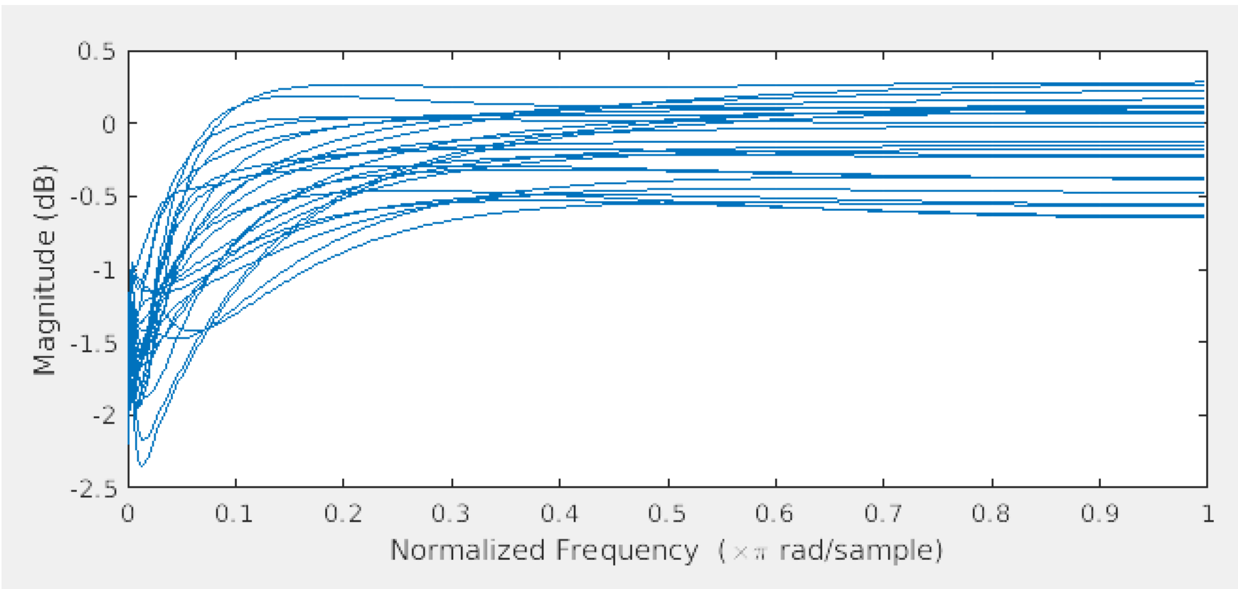

Figure 6: Frequency responses of filters based on glass absorption coefficients.

Additional copies of each impulse were placed to increase echo density in the late reverberations of each impulse; while this decision does not follow to the letter the rules of ISM, it makes a positive change to the sound of the impulse, as suggested in Smith (2005), and moreover, this loosening of timing and addition of further impulses could account for the effect of diffusion and scattering of reflected sound, something not accounted for elsewhere in the model.

3.5 Description of MATLAB files

The code for this project consists of several MATLAB scripts. The first script is `main.m`, which allows the user to set parameters such as building and street width and distance. It also allows the user to specify sampling rate and stereo width. `main.m` then calls `ir_generator.m` twice; due to the randomness discussed earlier, these responses are slightly different, though they share loudness curves that are about the same. Those two generated responses are mixed according to the specified stereo width, then written to a .wav file.

`ir_generator.m` is where the bulk of the logic is. This script calls `wall_filter.m`, a script that filters a given array based on absorption coefficients passed to it. To convert from absorption coefficient to biquad filter, this script calls `get_ls_coeffs.m`, `get_pf_coeffs.m`, and `get_hs_coeffs.m` to calculate the filter coefficients for low shelf, peaking, and high shelf filters, respectively. After calculating all impulses, the result in `ir_generator.m` is gently lowpassed at 1500 Hz and highpassed at 30 Hz before being written.

4. Analysis

Impulse responses for analysis were generated at 44100 samples per second, with building width set to 1000 feet, street width set to 20 feet, and distance set to 0, 2, and 4. The responses are generally characterized by large amounts of predelay, due to distances of thousands of feet from source to listener; 100-250 ms of ramp-up to the point of highest amplitude, and approximately exponential decay until a faster drop-off due to the geometry of the space. Shown here are waveforms and spectrograms for the IR with distance set to 4.

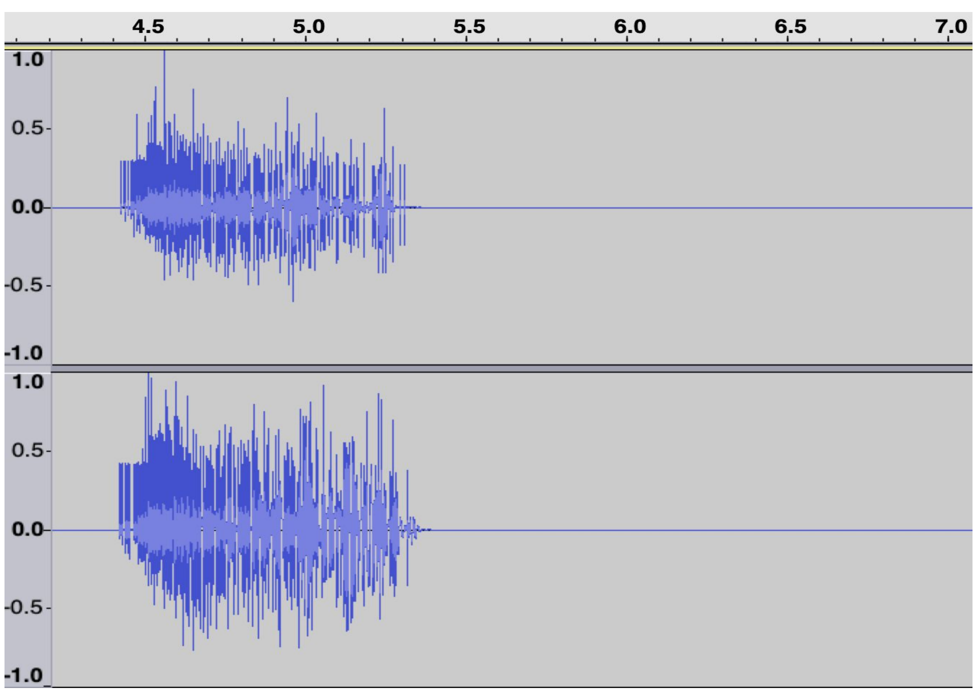

Figure 7: Impulse response waveform. Horizontal units: time (s). Vertical units: dBFS.

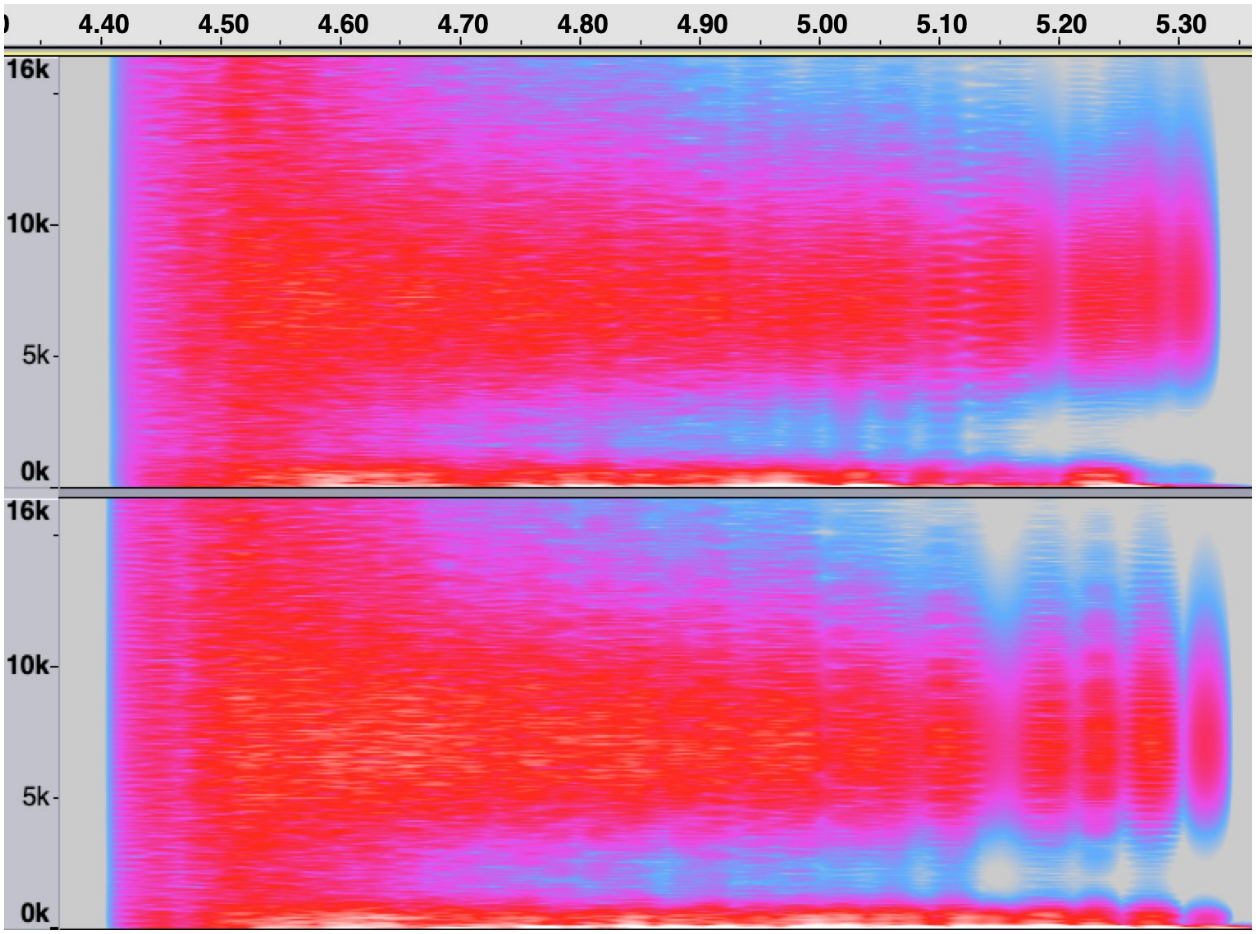

Figure 8: Impulse response spectrogram. Horizontal units: time (s). Vertical units: frequency (Hz).

Evident in the distribution of impulses in these responses is the geometry that they are based on. For instance, the IR with distance set to 2 has a shorter length (~50,000 samples) than the IR with distance set to 4 (~59,000 samples). One would expect this, since sound coming from farther away has more time to reflect and become more diffuse. Contrastingly, the IR generated with distance 0 has an especially long tail - since there are no alleyways to stop back-and-forth reflections, this tail would continue indefinitely if not for internal limits on the amount of VLs.

It can be seen from the spectrograms that these IRs contain a lot of low-frequency energy. This happened partially because later in the response, the density of valid paths decreased to a level where each impulse was audible and their frequency was uniform enough to be heard. Initially, this detracted from the usability of the IRs when placed in convolution reverb plugins and auditioned with different sounds (Smith (2005) recommends higher echo density in late reverberation for this very reason). Two things that helped a lot were the addition of random additional impulses near each original impulse as well as making the responses stereo rather than mono. That said, the existing echo density, while somewhat lower than recommended, offers a reverb sound that at the very least is different from more traditional or “ideal” reverberations and could be suitable for more experimental sounds. Moreover, adding additional amplitude envelopes to the IR via external plugins tames the late reverberations considerably.

5. Conclusion

Using the image source method, impulse responses are created based on the structure of cities. These IRs reflect the properties of the open spaces they were created in, sometimes functioning as a “blurred delay” rather than a more regular reverberator. Depending on the parameters used to generate them, they have a variety of characteristics, including different response lengths. Though there are multiple applications for these responses both within computer music and in more general digital mixing and music creation within the DAW, three specific uses are viewed at this point as most fruitful: using shorter-length IRs as “echos” or long delays; using shorter-length IRs as is, reminiscent of gated reverbs often used on drums; and using longer-length IRs as more traditional reverberators, tuning predelay and amplitude envelopes to create the desired sound.

Moving forward with this work, it is my intention to create an audio plugin that incorporates these algorithms. Rather than working with convolution, this plugin would use a circular buffer with taps calculated based on the math described earlier. Further work could also be done regarding simulation of diffuse reflection in the image source method, including development regarding number and propagation of additional impulses per virtual listener. Regardless, it is my hope that the sounds captured in these responses provide setting and inspiration for further acoustic and musical exploration of the sound of cities.

Acknowledgements

I would like to thank the following people: Jonathan Abel, for being my advisor for this project and allowing me to share in his knowledge of sound; Jonathan Berger, for being my academic advisor as a music major; the faculty, staff, and students of CCRMA, for constantly providing inspiration and guidance in music; and Anna Whittell, for creating the diagrams in this paper.

Works Cited

"Absorption coefficient chart". JCW Acoustic Supplies website. https://www.acoustic-supplies.com/absorption-coefficient-chart/. (June 2, 2020)

Asselineau, Marc. *Building Acoustics*. CRC Press, Boco Raton, 2015. 250 pp.

Bristow-Johnson, Robert, ed. Doug Schepers. "Cookbook formulae for audio equalizer biquad filter coefficients". https://www.w3.org/2011/audio/audio-eq-cookbook.html (June 2, 2020)

Collecchia, Regina E., *et al.* "On the Acoustics of Alleyways". Audio Engineering Society Convention Paper. 2014.

Engineering Acoustics. Wikibooks. https://en.wikibooks.org/wiki/Engineering_Acoustics (June 2, 2020)

Mehta, Prartha and Vaishalee Bhadradiya. "Measurement of Room Impulse Response Using Image Source Method", International Conference on Electrical, Electronics and Mechatronics, 2015

Smith, Julius O. *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects, December 2005 Edition,* online book at http://ccrma.stanford.edu/~jos/pasp05/, Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, Dec. 2005.