ECE 4271 Project 4

Birdsong - Crosswalk Audio Detection

Soohyun Kim
Joseph Spall IV

Introduction

Due to its patterned rhythm, the intermittent beeping sound produced by crosswalk devices can be of a great interference in recording and analyzing sound produced by birds. Thankfully, the crosswalk sound is easy to identify both by ear and algorithm; this report describes methodology used to identify and label this sound by algorithm.

Initial Research

Sample sounds were imported into the REAPER DAW and, using various plugins, its frequency spectrum observed. It was evident after several trials that the sound contained at least two peaks at ~880Hz and ~2700Hz. This was cross-verified with other samples; some showed a higher peak for ~880Hz, most showed higher for ~2700Hz, some both. No other peaks could be identified by the human eye, none before nor beyond the respective frequencies.

If then the 880Hz peak can be considered the fundamental, then the 2700Hz peak nicely aligns as the third harmonic of the Fourier series. Again, there was no discernible peak at the location of the second harmonic nor beyond, and this somewhat matches the sound's subjective description as being similar to a rectangular wave, which has nonzero harmonics at odd multiples of the fundamental (1, 3, 5, …). If the sound truly is a rectangular wave, then the fifth harmonic and beyond are too recessed for measure.

The period of the beeping sound was easily determined to be 1 second, and the duration of each beep >100ms. It became clear that the fixed frequency (1Hz) of the beep would be an important factor in identifying the sound.

A five-step daisy-chain of sound effect plugins within REAPER could be made to isolate the beeping, with good results:

1. RX 7 De-plosive by iZotope - Heavy wind noise is prevalent in all recordings, which is at least partly negated by this plugin.
2. RX 7 De-hum by iZotope - The fundamental frequency specified at 882.6 Hz and the number of harmonics to 3, the frequency content found there is amplified with adequate Q (71.1). The highpass filter is enabled at Fc = 683 Hz.
3. ReaFir by Cockos - The sound at the first and third harmonics are further amplified, the second killed.
4. RX 7 De-reverb by iZotope - At this stage the resulting sound is effective at isolating the beep, but undesirable frequency overlap from other sources of sound create a static background noise that resembles ringing. De-reverb is used to suppress this noise.
5. ReaGate by Cockos - a gate is set to dynamically filter the louder beeps from the static noise. An expander might be necessary prior to this step to heighten SNR.

This chain produces clean, isolated beeps as a result, given a good base SNR. It was used to assist the process of manually identifying sound files that contained the sound.

Methodology

Given the initial research into the crosswalk audio characteristics, we determined that more traditional signal processing methods could be utilized. Our program broke down into three main components as detailed in figure 1: foreground isolation, crosswalk isolation, pulse detection. Most of the audio processing was accomplished with the LibROSA audio analysis Python library. A majority of the filtration and feature detection occurs in the Short Time Fourier Transform (STFT) domain. The time sample is split up into small sections, with a FTT window size being 1024, and has a FFT applied to it, showing the frequency components present at different time sections. This proved effective in isolating aspects such as frequency, period, and duration of the signals.

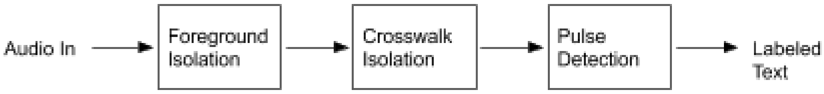


Figure 1. A diagram following the flow of the program

Foreground Isolation

The first step in the process was foreground isolation. There was extremely low signal to noise ratio in many of the audio clips, with inconsistent humming and white noise. Because of this noise, the first step was to isolate the important sounds in the clip, which is the “foreground” sound. This foreground isolation was accomplished with three main steps: take the magnitude of the STFT of the audio, perform a nearest neighbor decomposition filter generation in the STFT domain, and application of that filter using a soft mask.

The main portion of this step is the nearest neighbor decomposition filter. The process of the filter involves utilizing the cosine similarity for a comparison and then grouping similar frames (slices in time of the STFT) by taking the individual frequency bin median value. Additionally, the frames are separated by at least 2 seconds so that local continuous values do not bias the results. The minimum value of this filter and the original STFT is then calculated so that the filter is less than the input. The result of this then is softmasked with the original audio to get the foreground isolated, leaving the more distinct features including the crosswalk audio beeping. Most of this work is based off of the LibROSA tutorial for vocal isolation, with the results maintaining effective filtering. This is a very time consuming step (relative to the remainder of the processing) due to the searching and clustering involved. However, the impressive results that do not require prior knowledge of the noise are highlighted in the STFT spectrogram in figure 2 and audio wave representation in appendix figure 1.

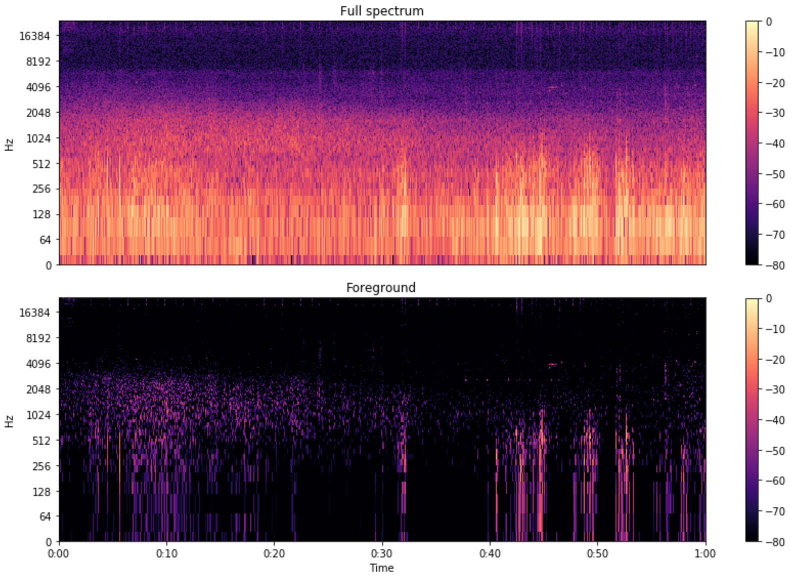


Figure 2. The STFT spectrogram of the original audio compared to just the foreground isolation. Crosswalk audio beeps are visible as the periodic dots in the 2680 Hz region in the 34 to 45 seconds.

Crosswalk Isolation

The next step is the actual crosswalk isolation. This is performed after the foreground isolation because of key properties of the crosswalk audio that are used as features for the isolation.

The first aspect is the frequency band isolation and thresholding. From the initial research, it was determined that the audio typically occurs in the 880 Hz and 2680 Hz frequency bands. The 880 Hz is often drowned out by surrounding noise, so the only focused band is the 2680 Hz band. This is implemented by setting all frequency components in the STFT magnitude not in the ranges of 2500 to 2800 Hz to zero. After this band filter, a very small thresholding occurs removing any components with less than -40 dB magnitude and assigning them to zero.

The next step is looking at the duration of the crosswalk audio beeps. They are relatively consistent, with the beginning of the beep to the end of the beep taking around 0.026 seconds. This time is converted to frames of the STFT. The frames are then grouped into continuous sets, which would indicate the beep is playing. The groups that are a minimum length of 0.026 seconds (in frames) are then considered long enough to be the beep as opposed to another feature or element of noise that slipped through the filter.

The next step is looking at the periodicity of the crosswalk audio beeps. Again, the period of the beeps is relatively consistent. The beginning of one beep to the beginning of the next beep is around 1 second. This is verified by calculating the center of each pulse, accomplished by averaging the beginning and end time of a pulse, and finding if any other pulses are 1 second away from it. Instances of beeps that are not periodic at 1 second interval are zeroed out. This concludes the crosswalk isolation steps, with the results after a final Inverse STFT being only the audible crosswalk beeps as seen in appendix figure 2.

Pulse Detection

The final step is the actual labeling. The results of the previous filtering steps leave an audio file that is mostly silent, with the remainder being instances of crosswalk audio beeps. This is accomplished with the LibROSA onset detector. The onset detector determines the peaks in the strength of the onset of sound. All default parameters are used besides the correct sampling rate in order for time estimation to be accurate and backtracking to the minimum of the strength detected for the start of the beep. The returned values are frames in LibROSA, so these are translated into seconds-based timestamps. These timestamps are then clustered to be about 1 second in duration to prevent periodic echoing triggering multiple times, highly reducing the false positive rate.

Final Result

The final classification is then complete. The times from the pulse detection are used as the starting time, and the observed duration of the beeps is used to calculate the ending time by simply adding to the started time. Figure 3 displays the results of the process over the audio waveform.

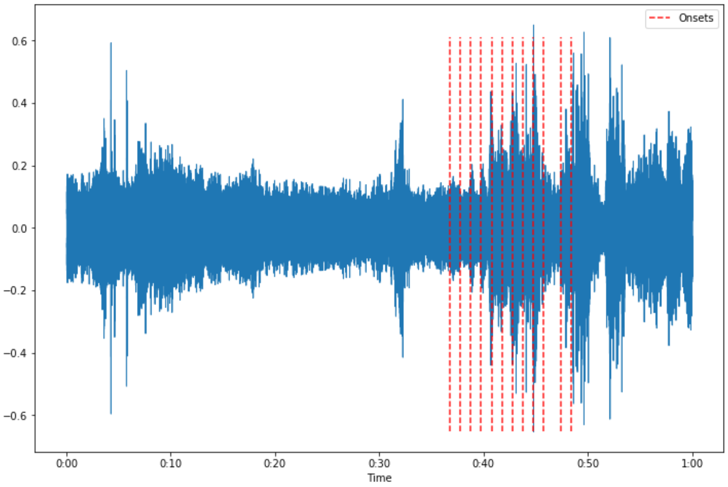


Figure 3. An example of an original audio waveform with the labeled crosswalk audio beeps.

Validation

Our process classifies actual beeps themselves to be the crosswalk audio, as opposed to the entire time frame from the starting of the first beep to the last beep in the sequence when the crosswalk audio is active. This “stricter” definition was still able to perform considerably well.

For comparison, seven files were manually hand-labelled by auditory and visual inspection. The five-step daisy-chain was used for assistance, but attempts at recording the sound directly to an empty track in the DAW was foiled by latency in the sound. Rendering the processed sound and re-importing it into the DAW was necessary, which then was used as a guide for finding timestamps of individual beeps; all in all, it was a very painstaking process. To alleviate some of the burden, the assumption of 1 second period and 100ms duration was made for all beeps.

After the hand labeling process, the final verification is also automated. Given the text output files of both the hand-labeled and the generated labels, the program generates ranges for each that a crosswalk is present defined as from the beginning to the end of the singular beep. A result is considered a true positive if for every hand labeled range there is an overlapping generated range. A result is considered a true negative if for every hand labeled range there is not an overlapping generated range. A result is considered a false positive if for any generated ranges there does not exist a corresponding hand labeled range.

There were two groups of audio files: average files and difficult files. For the average files, the crosswalk audio was clear to the human ear with standard quality headphones and no audio presets and low amounts of overpowering noise. The performance was 30 true positives, 7 false positives, and 4 true negatives.

For the difficult files, the crosswalk audio was often not discernable without high-quality headphones and extensive audio presents and/or had large amounts of overpowering noise. The performance was 12 true positives, 16 false positives, and 20 true negatives.

Run	Difficulty	True Positives	False Positives	True Negatives
1	Average	9	3	0
2	Average	11	4	0
3	Average	10	0	4
4	Difficult	6	7	16
5	Difficult	6	9	4

Table 1. Performance of algorithm compared to manually labeled output file.

Conclusion

Overall, we considered the performance of the algorithm pretty successful. Pretty consistently the algorithm correctly detects and labels the crosswalk audio pretty effectively. We made the system work on the substantially more difficult criteria of only labeling the actual beeps from the crosswalk audio as opposed to the entire duration of the beginning of the beep to the end of the last beep. Though performance declined for very difficult sound files, overwhelming noise in the clips would not be useful for bird song detection use as well. The algorithm was tuned for more false positives, which can be reduced if desired with included thresholding.

This original algorithm was the first approach tested, so the level of success prompted us to not need to attempt further approaches. In development, there were many versions that lacked certain components of the pipeline (e.g. not applying nearest-neighbor filter, not checking periodicity, etc.). One difference was the initial periodic detection approach. This feature originally was verified by taking the current center duration of the ranges and comparing only the previous and following. The two differences are then compared to the 1 second period, converted to frames, with one of them being required to be within a certain margin to be considered periodic. This was changed to the current method to prevent many false positives in a row, typically due to echo, throwing out an entire beep. However, the best performing pipeline obviously includes the components listed.

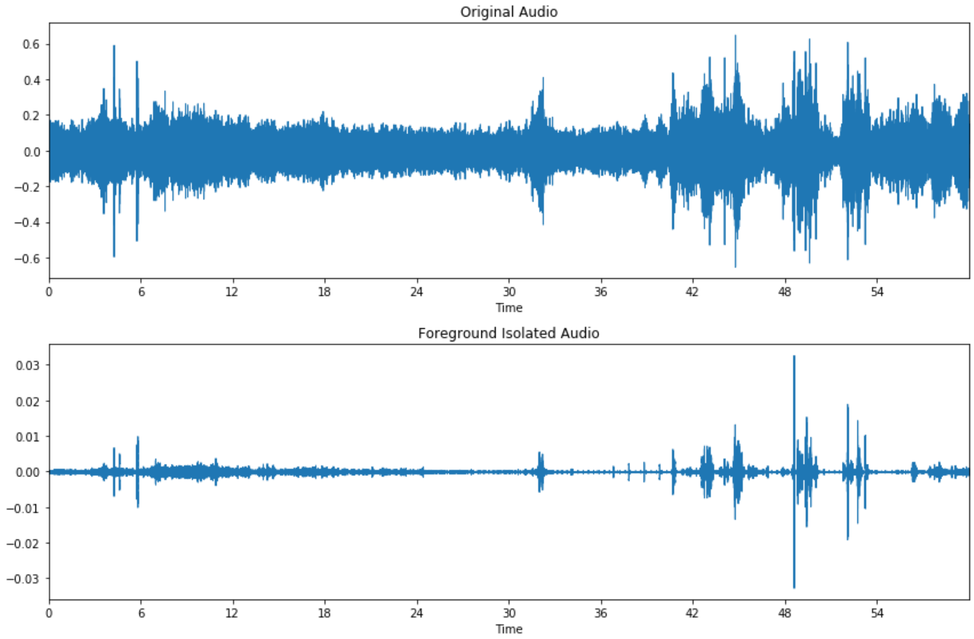
There are a few downsides to our algorithm. It is not generalized to all types of crosswalk audio sounds. However, given the use case on Georgia Tech's campus and consistent use of the same crosswalk audio sound throughout campus, this does not seem to be a problem. The algorithm also typically has trouble with very quiet crosswalk detection due to filtering that occurs. Many of the recordings have very low SNR where the crosswalk audio was detected, so this is to be expected. The last aspect is the requirement of the crosswalk beeps needing to be sequential and periodic as mentioned in the third component of crosswalk isolation methodology. The positives of reducing false positive detection seems to out-weight the commonality of single beeps being missed.

There are a couple upsides to our approach as opposed to utilizing strategies using neural networks or other machine learning approaches. There is not a need to label large datasets for examples, which can be a very boring and time consuming task. Additionally, this method is effective at picking up instances that are hard to hear in the audio clips, which labeled data may not include. Finally, a larger variety of training data would be required that would not necessarily correlate to human comprehensible parameters as opposed to tunable parameters present in our algorithm.

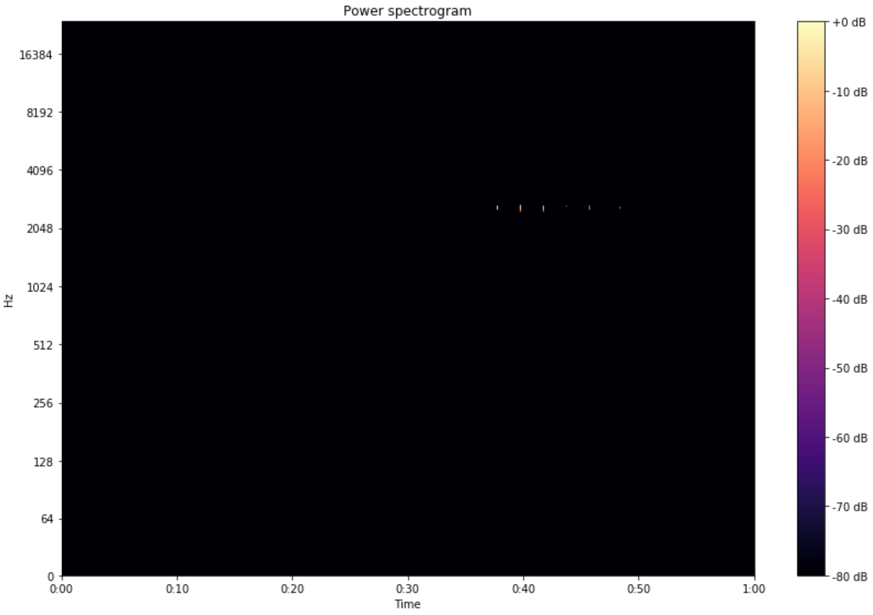
Work Distribution

Soohyun was responsible for the following: identifying and analyzing suitable properties of the crosswalk sound for algorithmic processing, both frequency-based and dynamics-based; managing and hand-labelling audio files to use as test cases
Joseph was responsible for the following: all programming work including the labeler and verification, the entire "Methodology" section, and parts of the "Validation" and "Conclusion" sections.

Appendix



Appendix Figure 1. The audio waveform plot of the original audio compared to just the foreground isolation. The actual crosswalk audio is not immediately discernible, but noise reduction is clearly present.



Appendix Figure 2. The STFT spectrogram of the final crosswalk audio, with color dots being the remaining crosswalk components.