Chromazam

Song Identification using Chromagram

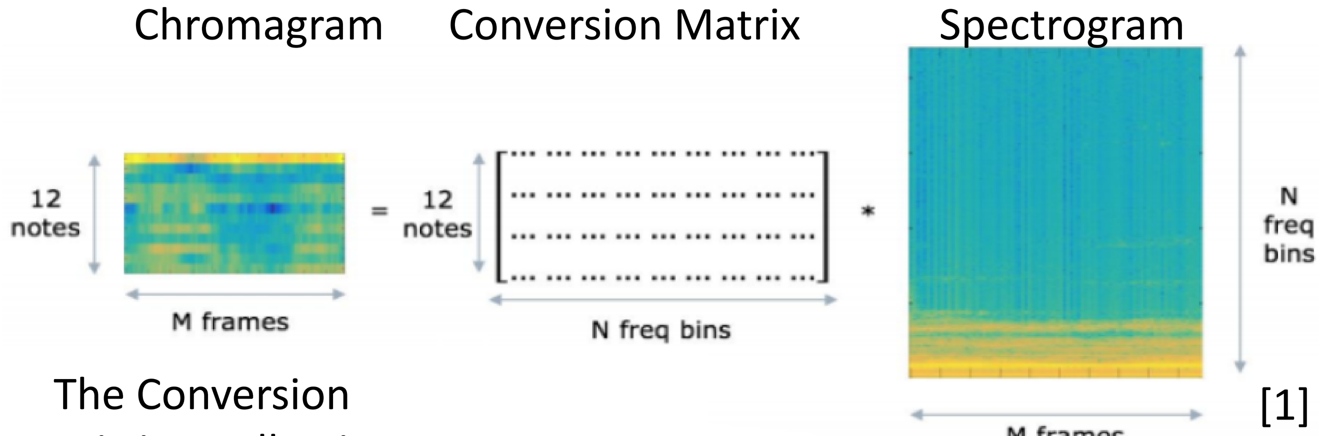
Steven Belitzky, Christopher Palace, Albert Peyton

Abstract

We designed an algorithm like that of the popular app "Shazam"; upon recording a 5 to 20 second clip from anywhere in a song, our algorithm will accurately identify the song from our database. In contrast to the "Shazam" algorithm, which uses spectrograms for song identification, our algorithm uses Chromagrams to accurately identify songs. Our method could provide some advantages over the "Shazam" algorithm such as being able to identify a song regardless of the key and without the recording which is saved in our database. Upon completion of our program, we will test how accurate our algorithm can identify a song of variable input length and distortion to compare its correctness to the "Shazam" method.

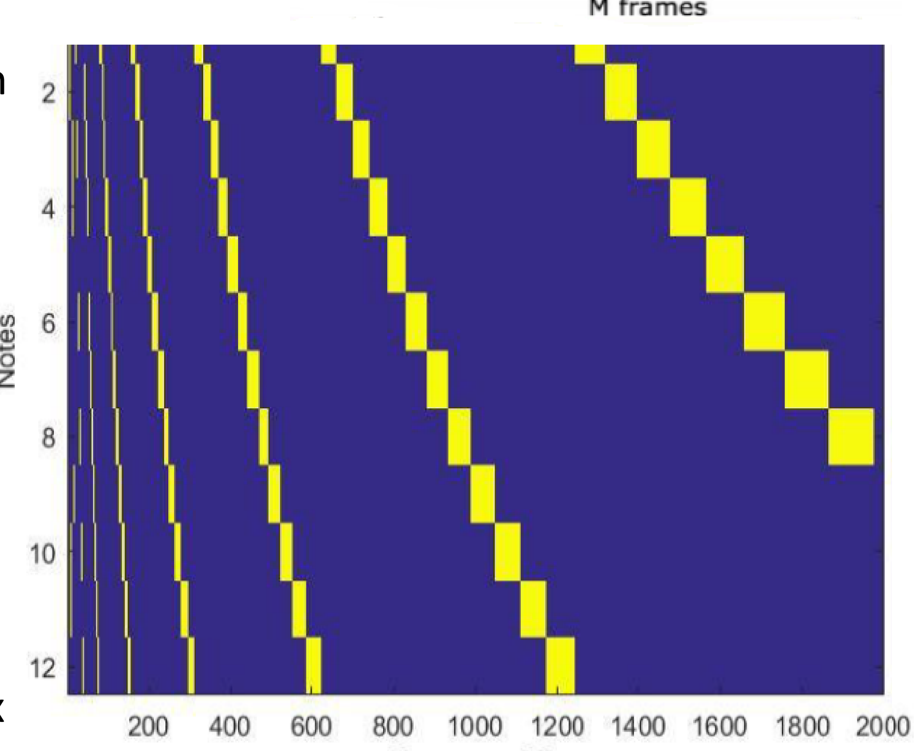
Background

A Chromagram is a condensed form of the spectral information of a given waveform. It converts the different frequency responses across all audible octive bands into their pitch classes: their note namesgiven by western music theory over time. As seen in the figure below, the Chromagram is obtained by taking the spectrogram of a signal, and using matrix multiplication with said spectrogram a conversion matrix.



[1]

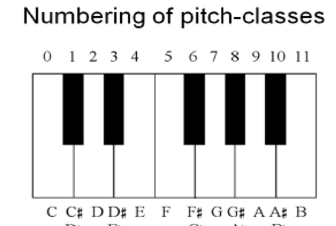
The Conversion matrix is a collection of 1s and 0s that span the entirety of range of human hearing. The 1's represent the frequencies at which each note occurs, whilst the 0's represent the space x between them. By using matrix multiplication with the conversion matrix and the spectrogram, the result is a 12xM matrix, where M represents the time domain. The Conversion matrix pictured above



Methods

Preprocessing

- Take the Chromagram of each song in the database
- The largest value (most present note) is saved at every sample
- The difference between each value and its successive value is calculated
- If values are negative, 12 is added to them
- The input is now represented by an array of integers between 0 and 11



Recording input

- Input (5-20 second song sample)

Preprocessing input

- Preprocessing is run on the recorded input

Processing

- Euclidean distance is calculated between the recorded input and the beginning of a song in the database
- The input is moved one sample forward in the database song and the above step is repeated.
- The smallest Euclidean distance value is saved this song.
- The previous steps repeat for each song

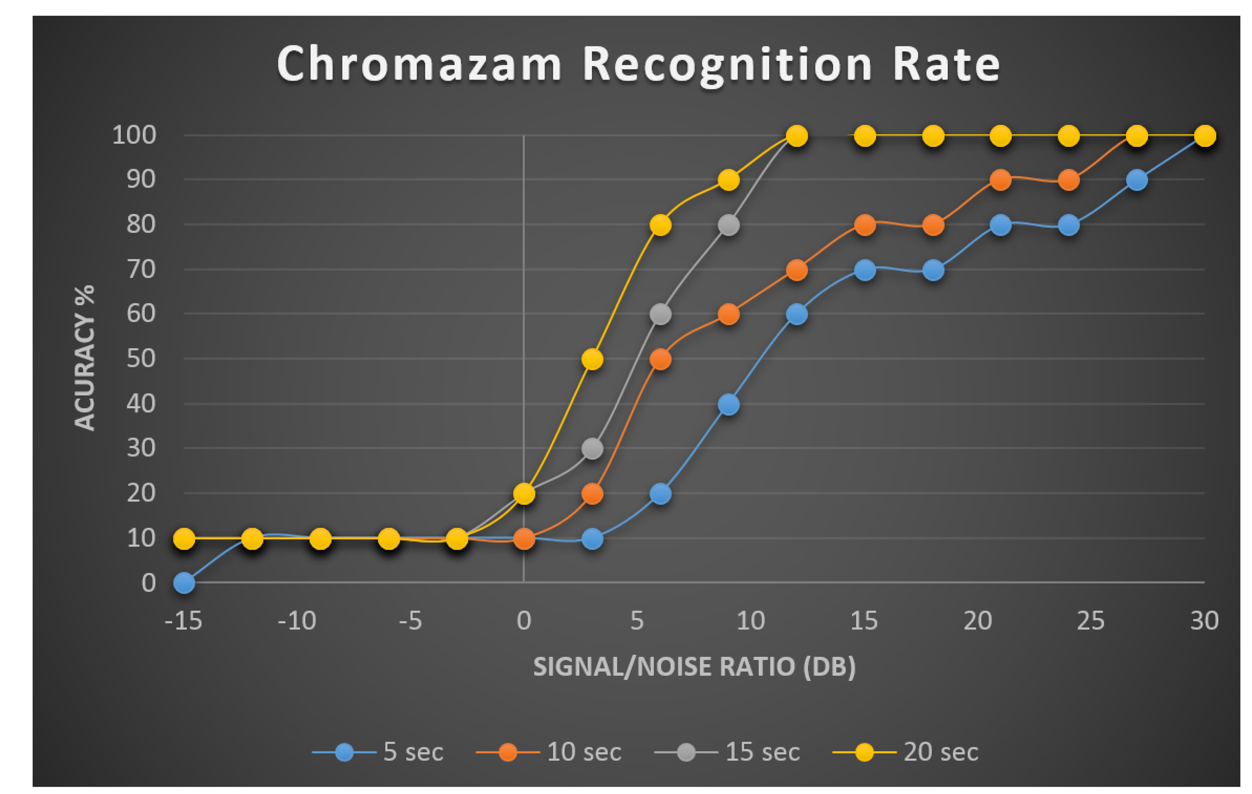
$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

Identification

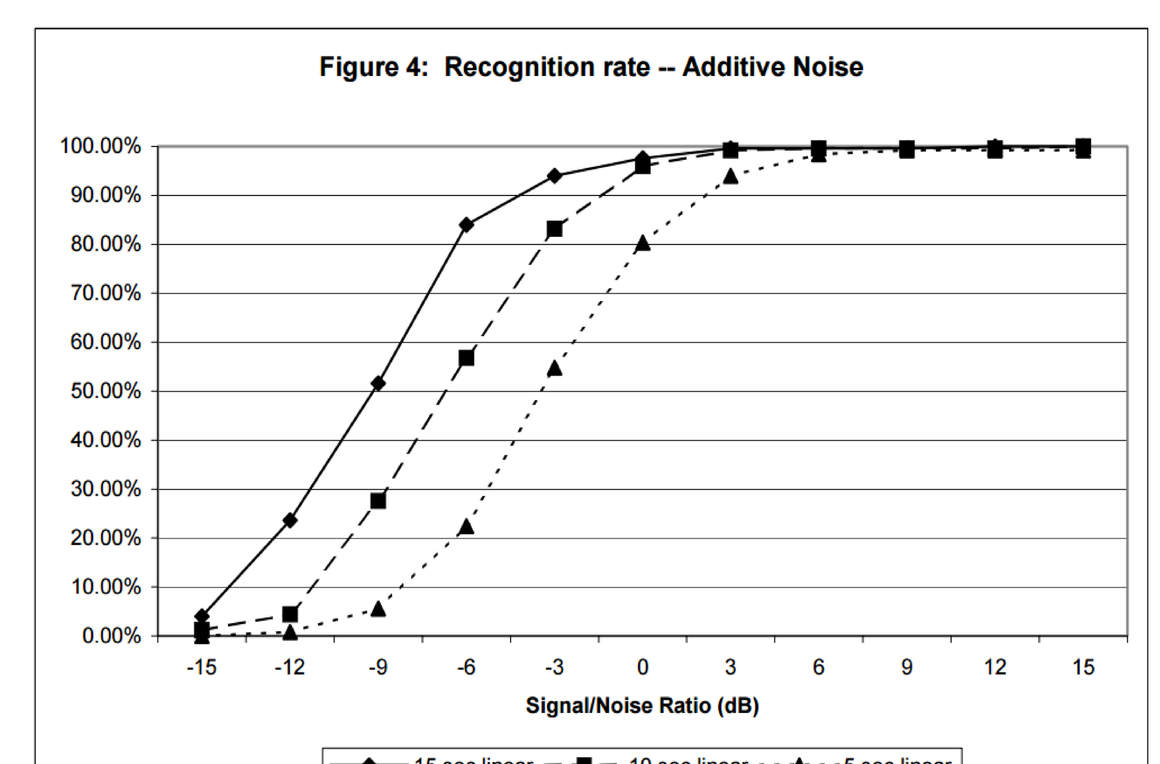
- The song corresponding to the smallest Euclidean distance value is returned

Tests

Initially, we tested our Chromazam algorithm with 5 complex polyphonic songs, consisting of multiple instruments and various melodies and rhythmic patterns. This test yielded a 0% rate of accuracy. Upon seeing these results, we compiled a database of 10 simpler songs, that are more less monophonic, such as solo violin, tuba, vocal and trombone pieces. With these simple songs our algorithm was quite effective. We tested the accuracy of our algorithm with varying lengths of recorded signal, and by adding different ammounts of noise to the recorded signal.



Below are the results from a similar test run by Shazam on the Shazam Algorithym. As you can see our Algorithym is less effective, but still is usefull for song identification.



[2]

Conclusions

Our current algorithm is not comparable to Shazam's because it does not have the ability to accurately identify polyphonic songs. When multiple melodic or harmonic lines are present in the inputed waveform, our program is overwhelmed by the amount of activity in the spectrum and is unable to distinguish the most prevelent pitch classes. That being said, we were successful at creating an algorithm that can successfully and efficiently identify monophonic songs from a database using a chromagram. We were also successful at identifying a song regardless of its key, something the Shazam app does not allow. It conclusion, we were quite overzelous in our proposal and were under the impression that we would be able to accomplish a lot more than we did, but in most ways our project was a success and we have still have the ability to develop and further improve our algorithm.

Future Work

For our future work with this project we plant to implement a source separation algorithm which will be able to extract different parts from a piece to be analyzed. These parts could be anything; the vocals, lead guitar, piano melody or really any melodic part which will be isolated and then analyzed opposed analysising the waveform of the song as a whole. This will solve the problem of extraneous information overwhelming the processing. When this problem is solved and our algorithm will be able to account for polyphonic pieces, we will be able to increase the number of songs in our database and construct hash tables to store data and run our program as efficiently and effectively as possible.

References

[1]Y. Zhang, "Chromagram Representation for Musical Signals", University of Rochester, 2017.

[2]Shazam Entertainment, Ltd, "An Industrial-Strength Audio Search Algorithm", 2003.

Contact Us

Steven Belitzky - sbelitzk@u.rochester.edu
Christopher Palace - cpalace@u.rochester.edu
Albert Peyton - apeyton@u.rochester.edu