Comparative Analysis of an Armenian Hymn Through Digital Signal Processing and Music Information Retrieval

by

Jack Bouhanna

S.B. Electrical Science and Engineering, Music
Massachusetts Institute of Technology, 2021

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 2022

Signature of Author:

____

Department of Electrical Engineering and Computer Science
May 9th, 2022

Certified by:

____

Garo Saraydarian
Lecturer of Music
Thesis Co-Supervisor

Certified by:

____

Peter Hagelstein
Associate Professor of Electrical Engineering
Thesis Co-Supervisor

Accepted by:

Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Comparative Analysis of an Armenian Hymn Through Digital Signal Processing and Music Information Retrieval

by

Jack Bouhanna

Submitted to the Department of Electrical Engineering and Computer Science on May 9, 2022 in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Armenian music has existed for centuries, dating back to several millennia BC. The music has undoubtedly evolved over time, whether passed down traditionally or through reimaginations of the original piece. Despite straying from the original versions, the music nonetheless keeps the spirit and tradition behind them intact.

This thesis will compare and analyze the harmonic differences in a famous Armenian Hymn, Տէր Ողորմեա ("Der Voghormia", meaning "Lord Have Mercy"). The baseline version will be the one that is found in the 20th-century manuscript written by Gomidas Vartabed, and will be compared against later renditions. This will be performed by using several different techniques and algorithms from the Digital Signal Processing (DSP) and Music Information Retrieval (MIR) fields. The final products will be implemented through Python programming, along with related helper packages and toolkits.

Thesis Co-Supervisor: Garo Saraydarian
Title: Lecturer of Music

Thesis Co-Supervisor: Peter Hagelstein
Title: Associate Professor of Electrical Engineering

Acknowledgements

I would like to thank my two co-advisors, Garo Saraydarian and Peter Hagelstein, for their support, guidance, encouragement, and providing me with the opportunity to dive into a topic that combines my passions of engineering and music. I would like to thank Eran Egozy for introducing me to this intersection of computation, signals, and music. I would like to thank my friends for their support and for helping me escape stress. Finally, I would especially like to thank my family. Their continued love and care has been monumental to my success both in and out of MIT.

A special thank you goes out to my mother Sosi, my cousin Vazrig, my grandmother Armenouhi, and my great aunt Sona. Despite my late interest in my heritage, their guidance, patience, and teaching has encouraged me to continue learning and embrace my culture.

Contents

Chapter 1: Introduction

Music has always been a fundamental part of society. Stretching back many thousands of years, music served as a way to entertain others, exchange ideas, and establish identities. Today, that still holds true. Millions of people worldwide interact with music due to its uses in mass media, entertainment, and social activities. The way music makes its way into our lives underwent a radical change due to advancements in digital technology. In addition to the mass creation and distribution of music through digital means, we can also computationally study properties of music.

Computational Music Analysis incorporates techniques and tools from both digital signal processing (DSP) and music information retrieval (MIR). The goal of MIR is to extract interesting data points about a piece of music in question, such as its tempo, structure, chords, and instrumentation. All of this information and more can be extracted from a digital audio signal, incorporating techniques from DSP. The key part of DSP is utilizing the discrete Fourier transform, which decomposes functions of time into functions of frequency. This transform helps dissect what frequencies with what magnitudes are present in an audio signal, allowing us to extract musical features to analyze.

Armenian music has been around for hundreds and thousands of years. However, two major events in Armenian history significantly impacted the survival of Armenian culture: the Armenian Genocide in 1915, and the establishment of Soviet Armenia in 1922. As a result, the way Armenian music has been studied, written, and performed has changed due to the destruction of musical resources, change in political ruling class, and change in geography. Musical traditions and manuscripts that survived may be different from what once was, and the music of the Armenian Diaspora may stray from the styles and methods used in Armenia today. In addition, the development of digital technology has forever shaped the future of music. Between new instruments, analysis tools, and audio formats, there is a lot more that is possible with music today than there ever was before.

The hymn we wish to analyze is Gomidas Vartabed's composition of “Der Voghormia”. The analysis will involve designing algorithms that will allow us to compare various renditions of the piece, and see how similar or different these different versions are from Gomidas's. The necessary DSP and MIR methods will be implemented using the Python programming language. With the exception of some helper functions to simplify some math and plotting, all of the code will be written from scratch. The algorithms will work off of WAV audio files, which can be generated and exported through the score writing program MuseScore and the audio editing software Audacity. Figures (Fig.) and equations (Eq.) will be labeled by their corresponding chapter and subsection. Specific songs will be referred to by their names from Appendix A.

Chapter 2: Armenian Music Research

Before going into the details of the algorithms, it is important to understand how Armenian music in general can be different from what we are used to seeing and hearing in the West. A twelve tone scale is common, but is not the standard around the world. Armenian music does have its similarities with Western music, but there are areas where they differ.

2.1 Armenian Music System

A complication considered in implementing the necessary algorithms and parameters is the structure of Armenian music. Notable differences include its scalar structure, instrumentation, and harmonization. Although we are dealing with a hymn that is purely vocal, covers and remixes that exist due to modern music technology may contribute additional instrumentation and ornamentation to the music. Armenian music does not follow the same structure as Western music, where there is an established system of 12 tones. The structure of Armenian music originates with its chant music, similarly to Western music with Gregorian chant. Armenian chant music adheres to a system known as *octoechoes* (Kerovpyan and Kerovpyan, 2020). This system contains a specific subset of liturgical music known as *sharagan*, which made use of a tetrachordal system with non-Western intervals. These intervals include major and minor semitones, subminor thirds, and medium thirds. As a result, attention to the intervals, notation, and harmony used in “Der Voghormia" is critical and should be considered when coding the necessary algorithms. This is especially due to how MIDI, a digital interface for audio, typically calculates the frequency of a note. MIDI notes take on a value from 0 to 127, and are converted to a frequency based off of the twelve-tone scale by:

$$f(n) = 440 \times 2^{\frac{n-69}{12}} \qquad \text{(Eq. 2.1.1)}$$

The frequency is in hertz, 440 Hz is the common frequency used for A4 (the “A” above middle C), and 69 is the MIDI number for A4. A4 is often used as the reference pitch for the conversion. If any versions of “Der Voghormia” use pitches outside of the Western twelve-tone scale, this formula would not work and our algorithms must be modified. Unless specified otherwise, we will assume the 12-TET system, as the exact verse we are comparing against uses a 12-tone scale.

2.2 Gomidas Vartabed

Composing, arranging, and transcribing over 3,000 works of folk music, Gomidas Vartabed's (1869 - 1935) works received praise from local communities and abroad (Vagramian-Nishanian, 1981). Although his career began in the Caucasus, he attended many opportunities in Europe which established the link between Armenian music and Western music. His studies helped bring his Eastern influences, such as instruments, to the West. Western influences, such as the notation system, were also brought back to the East. While abroad, he entertained professional journals with his insight on eastern music. These insights were not only on the origins and meanings behind Armenian music, but also the Arabic, Kurdish, Turkish, and Persian melodies that he studied as part of his ethnomusicological career.

2.3 Der Voghormia

"Der Voghormia" was first composed by 18th-century composer Catholicos Simeon Yerevantzi (1710 - 1780), also known as Simeon I of Yerevan (Capan, 2005). Simeon was known for establishing the first printing press in Armenia, embracing the production of manuscripts and books. The hymn embodies sorrow, joy, and hope, calling to the Lord to have mercy. It is often performed before Holy Communion, where the body and blood of Christ is received. The song has been performed in churches and at events for hundreds of years, and continues to have a significant emotional and spiritual impact for many.

The manuscript by Gomidas contains four voices: three tenors with one being solo, and a bass. There are several verses within the piece, but analysis will focus primarily on the structure of the first verse, as it dictates the pattern for the rest of the piece. Although the melodies are mostly the same, the version by Gomidas is the most famous. This makes it our prime candidate to compare other versions to it rather than the version by Catholicos Simeon. Both manuscripts can be found in Appendix B.

Chapter 3: Related Works

There are many research papers and theses covering related ideas out there. The International Audio Labs Erlangen has many theses and research papers that implement various analysis topics and techniques for the purposes of computational music analysis (Grosche, 2012). The overlapping topics between these references and our analysis are pitch and chroma based features, dynamic time warping, audio segmentation, and chord recognition. Another related topic is audio segmentation through harmonic-percussive separation, where the harmonic and percussive components of a song are extracted into their own signals (Driedger, 2016). These topics cover a wide area of analysis, some of which will be explored in-depth in the following sections.

One thesis on Georgian vocal music looks into the ideas of audio segmentation and fundamental frequency estimation in monophonic and polyphonic songs (Rosenzweig, 2017). These serve to identify the various voices in these songs, and then extract what pitches are being used. Another honors thesis from Wheaton College uses hidden Markov models to measure the similarity between western classical piano pieces (Liu, 2019). This focused on parameter estimation using an expectation-maximization algorithm, and defining a similarity metric from it.

The ideas above are just the tip of the iceberg. The content covered in this thesis is not the ultimate approach to achieve these goals, but serves as a showcase as to what computational music analysis methods exist and how they are established.

Chapter 4: The Fourier Transform

The Fourier transform, generally speaking, is an integral transform that decomposes a function of time or space into a function of temporal or spatial frequency. In simpler terms, consider a musical chord. The Fourier transform would decompose that chord into pitches with their corresponding intensities. This application is significant to the mathematical and computational exploration of audio. Specifically, we will be utilizing the **discrete Fourier transform (DFT)** and its relative, the **short-time Fourier transform (STFT)**, to showcase our analysis.

4.1 The Discrete Fourier Transform

Mathematically speaking, the DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{2\pi jkn}{N}} \qquad \text{(Eq. 4.1.1)}$$

N is the length of our time sequence ***x*[*n*]**, ***k*** is the frequency index, and ***j*** is the imaginary unit. Our output ***X*[*k*]** is a frequency sequence also of length ***N***.

This takes a time sequence of complex numbers and transforms them into a frequency sequence of complex numbers. Given a sequence of music, the DFT allows us to transform that sequence into its constituent frequencies, which opens the door to the analysis of these frequencies. The DFT is also an invertible process, meaning we can go back and forth between the time and frequency domains.

To translate our frequency index to a numerical frequency value in Hertz, we can use the relationship:

$$f(k) = \frac{kF_s}{N} \qquad \text{(Eq. 4.1.2)}$$

Where ***Fs*** is the sampling rate of our audio - we will be using 22050 Hz. This can tell us what frequencies are present in a song as long as we know the sampling rate and length of the original sequence.

If we know the length of our song is ***T*** seconds, then we can find that the time corresponding to a certain sample ***n*** is:

$$t(n) = \frac{nT}{F_s} \qquad \text{(Eq. 4.1.3)}$$

4.2 The Short-Time Fourier Transform

Related to the DFT is the STFT, which tracks the magnitude and phase content of a signal. The STFT works as a DFT that is windowed at various steps. The goal is to consider short sections of the signal to extract exactly when certain frequencies occur, as the DFT's frequency information is averaged over the entire time domain.

The STFT is defined as:

$$\mathcal{X}[k,m] = \sum_{n=0}^{N-1} x[n+mH]w[n]e^{-j2\pi kn/N} \qquad \text{(Eq. 4.2.1)}$$

Similarly to the DFT, we have that ***N*** is the length of our time sequence ***x*[*n*]**, ***k*** is the frequency index, and ***j*** is the imaginary unit. For the STFT, ***k*** will be a value ranging from 0 to ***N***/2 + 1.

Our windowing function ***w*[*n*]** is also of length ***N***. We want the whole signal captured, so that a decay can be applied to samples outside of our short sections. The Hann window is often chosen due to its good tradeoff between time resolution and frequency. Since we are working with audio signals, the band of frequencies we obtain over time works better with a "smoothing" window relative to, say, a rectangular window. These windows are separated by our hop size, ***H***. The variable ***m*** is an integer value ranging from 0 to ***M*** - 1, where ***M*** is the total number of hops.

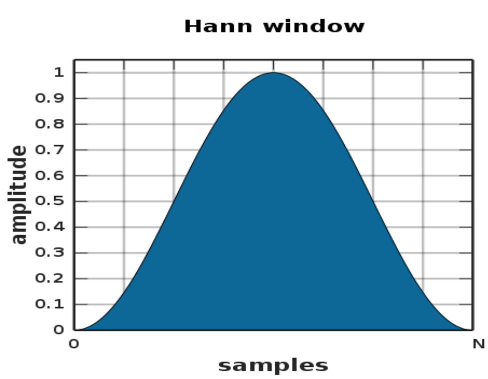

Fig. 4.2.1, Hann window

Just like with the DFT, we can invert the STFT to get the inverse STFT, or iSTFT. The iSTFT will come into play in chapter 5.4.

Frequency is translated just like how it was for the DFT. We can also recover the time value (in seconds) of each column. The time value of each column given by:

$$t(m) = \frac{mH}{F_s} \qquad \text{(Eq. 4.2.2)}$$

Chapter 5: Fourier Applications and Methods

Since the Fourier transform gives us access to frequency information, we can utilize it to uncover even more properties of an audio signal. This can be done by visualizing our audio (chapter 5.1). Building off of that, we can obtain the information to perform our comparisons and analysis. We first need to determine which sections of music match (chapter 5.2). Then, we need to extract the relevant harmonic data (chapter 5.3). Finally, we will define and detail a quantitative way to compare our songs (chapter 5.4).

5.1 Spectrograms and Chromagrams

The spectrogram is a picture of sound, displaying frequency against time. It is defined as the square of the magnitude of the STFT:

$$\text{Spectrogram}(k, m) = |\mathcal{X}[k, m]|^2 \qquad \text{(Eq. 5.1.1)}$$

The spectrogram of Der_Voghormia_1.wav, our baseline song, is shown below.

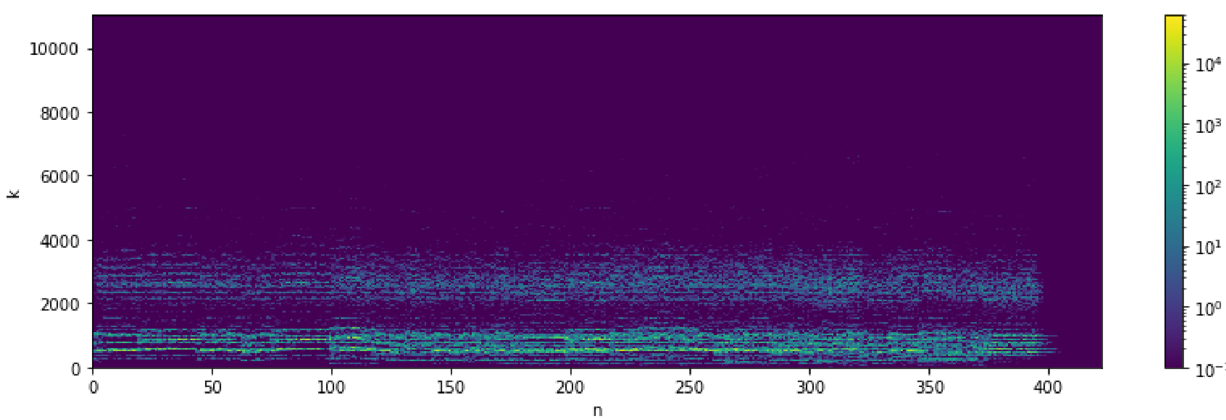

Fig. 5.1.1, spectrogram of Der_Voghormia_1

As mentioned before, the piece consists of three tenor voices and a base voice. We can see the more intense areas are towards the lower ***k*** values, which supports this idea. In general for live audio, intensities in higher frequencies can appear due to overtones.

Built off of the spectrogram is the **chromagram**, which groups notes by pitch class rather than frequency. This is independent of octave and timbre, meaning the chromagram is a very useful tool in analyzing melody and harmony. We will again assume a twelve-tone equal temperament scale, but note that modifications may be necessary depending on the music used for analysis.

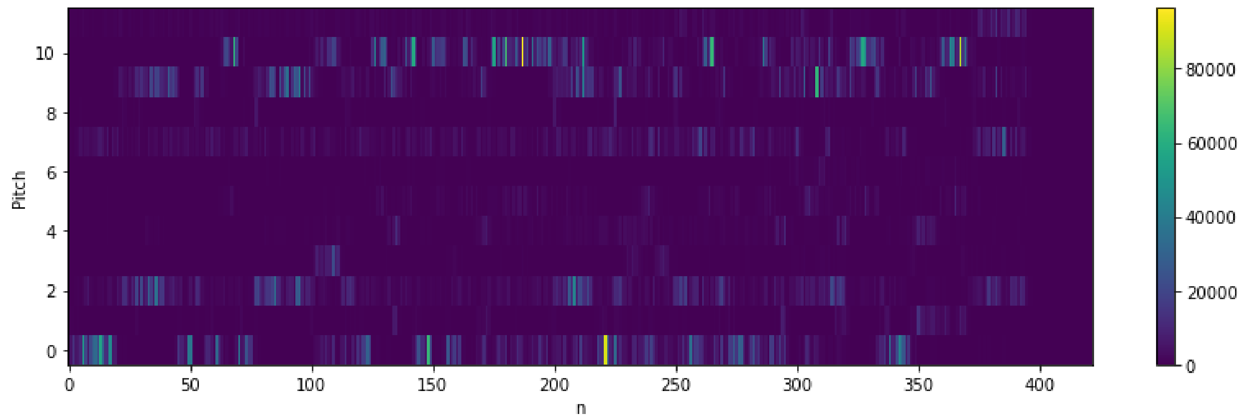

Fig. 5.1.2, Chromagram of Der_Voghormia_1

The chromagram is generated through matrix manipulation of the spectrogram. Since we hear pitch proportional to the logarithm of frequency, we first convert the spectrogram into a log-frequency spectrogram. This spectrogram assigns frequency bins to a pitch, making use of the relationship described by (Eq. 4.1.2). From there, we use a conversion matrix to combine a note and its octaves into a pitch class. This is to measure which notes show up, rather than segregating them by pitch class. This conversion matrix is 12-by-128, representing the 128 MIDI pitches and 12 tone scale (with 0 representing “C”, and each consecutive integer representing a half-step up).

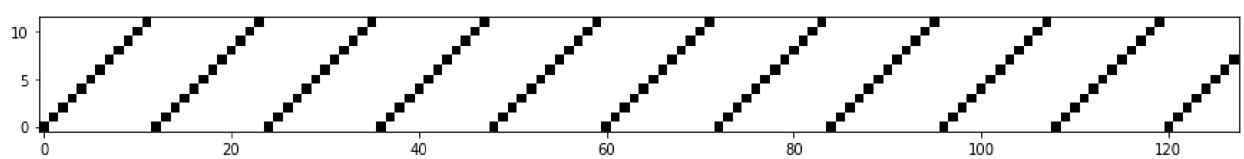

Fig. 5.1.3, Spec-to-Chroma conversion matrix

The multiplication of this conversion matrix with the log-frequency spectrogram results in a chromagram. However, there are several modifications we can make to the chromagram to enhance its properties. One of the most basic is applying spectral smoothing to the chromagram by means of logarithmic compression. This compression helps accentuate the lower energy notes. For our chromagram C, we can apply a compression factor γ that multiplies every value in our chromagram such that our new chromagram is equal to:

$$C_\gamma = \log(1 + \gamma \cdot C) \quad \text{(Eq. 5.1.2)}$$

The results of compressing by a factor of 1 is shown below. The note occurrences and their intensities (watch the color scale!) are much more clear than before:

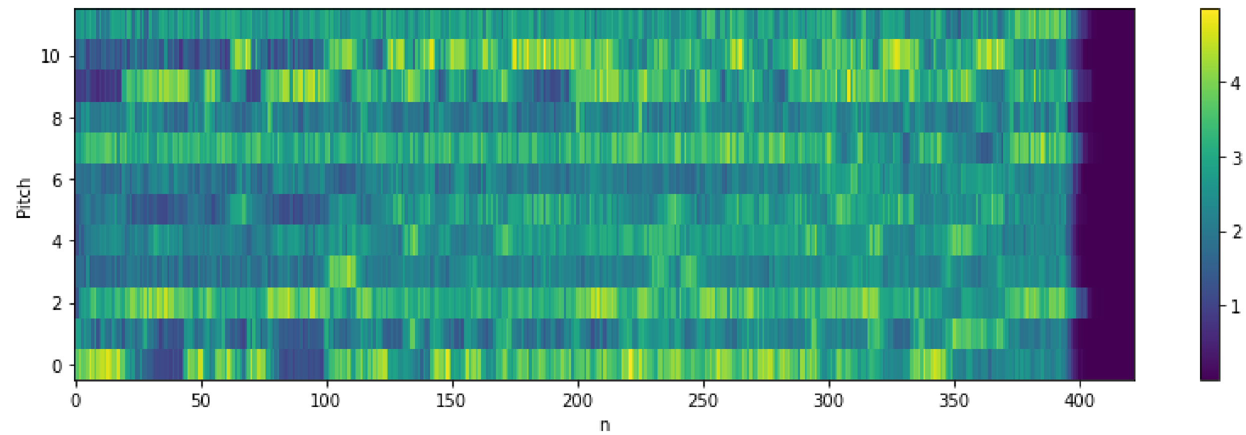

Fig. 5.1.4, Log10 compression of Der_Voghormia_1 with γ = 1

Another technique known as CENS (Chroma Energy Normalized Statistics) uses five modifications to the chromagram in order to suppress noise, ignore differences in volume, and clean up any small fluctuations (Müller, 2015, p. 375). This is done by first normalizing with the Manhattan norm, then performing logarithmic quantization. From there, we time-smooth and downsample the chromagram, which is finally normalized again by the Euclidean norm.

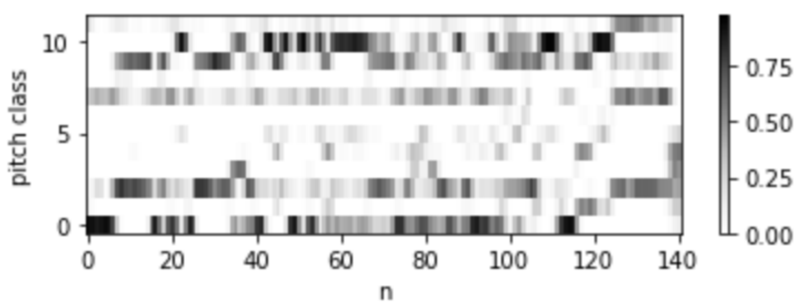

Fig. 5.1.5, CENS Chromagram of Der_Voghormia_1

Spectrograms and chromagrams will be vital to the following algorithms and analysis, and can be optimized further to improve the quality of our analysis. These optimizations and enhancements will be discussed in later sections, after demonstrating how these ideas serve as building blocks for our algorithms.

5.2 Dynamic Time Warping

Chroma representations of music allow us to compare and contrast the features in a song. These features can be synchronized into temporal correspondence through **Dynamic Time Warping (DTW)**, which finds an optimal alignment between two time-dependent sequences (Müller, 2015, p. 131). In our case, these sequences are chroma vectors. Given two different versions of a song, we can use DTW to scan for a section of music in one song to find the corresponding section of music in the other. Say there is a section of our baseline “Der Voghormia” that is 5 to 10 seconds into the piece that we wish to find in another version. If we were to use DTW with a version of “Der Voghormia” that is played twice as slow, we would receive an output that tells us that the corresponding section takes place in the 10 to 20 second frame of this slower piece.

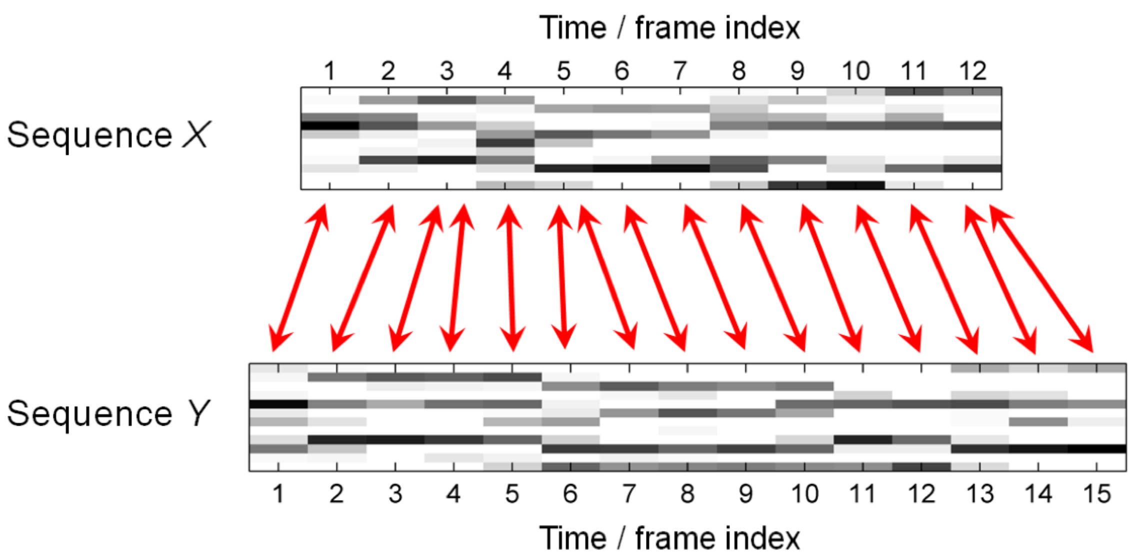

Fig. 5.2.1, How DTW aligns two sequences (Müller, 2015, p. 132)

In order to perform DTW, we must first generate an optimized **cost matrix** between the two chroma vectors **x** of size **N**, and **y** of size **M**. The cost matrix **C** compares every location of one chromagram to every location of the other chromagram. The cost is based on the cosine distance between the chroma vectors, namely:

$$\mathbf{C}[n,m] = 1 - \frac{\langle x_n, y_m \rangle}{\|x_n\| \cdot \|y_m\|} \qquad \text{(Eq. 5.2.1)}$$

Where ***n, m*** take on integer values from 0 to **N, M** respectively.

Other metrics such as Euclidean distance may be used, but cosine distance is mainly chosen because it is calculated independently of vector length. Instead, it only considers the twelve chroma bands and their respective energy distributions, which also makes it fast to compute. Note that for normalized chromagrams, the denominator is equal to 1.

Based on the cost matrix, we can compute a number of paths **P** that get us from the bottom-left of our matrix (i.e., the start of our songs) to the top-right of our matrix (i.e., the ends of our songs). However, our goal is to find the optimal path from start to finish. We can find this optimal path **P*** by first creating an accumulated cost matrix **D. D** calculates the optimal costs of all subpaths from start to finish for our two chroma vectors by choosing the adjacent preceding cell with the lowest cost. From there, we generate a backtracking matrix **B** that serves as a pointer to the matrix cells with the least cost. The optimal path is then found by starting at the top right of **B**, and following the pointers to the bottom left. An example of this process in action is depicted below:

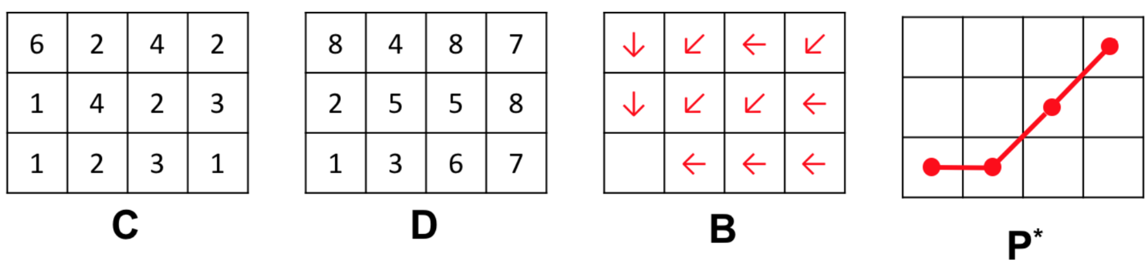

Fig. 5.2.2, Example of a full DTW process

DTW performed on Der_Voghormia_1.wav and Der_Voghormia_2.wav is plotted below, with the optimal path in red. Note the difference in song lengths - the optimal path reveals the alignment between the two songs, allowing us to find where an excerpt in one song can be found in the other:

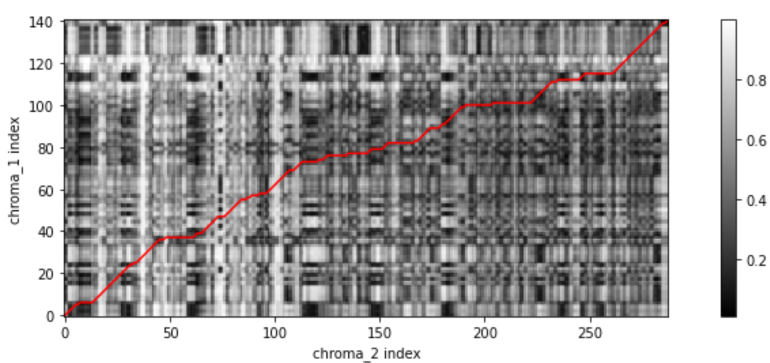

Fig. 5.2.3, DTW of CENS chromas between Der_Voghormia_1.wav and Der_Voghormia_2.wav.

5.3 Audio Decomposition

Der_Voghormia_1.wav is purely vocal, but what if we are comparing it to a version that has percussive parts? Percussion would influence the spectrogram and would impact our analysis. Thankfully, the spectrogram can tell us whether we are having a tonal sound or a sharp impulse. This is prominent in Der_Voghormia_2.wav, which features some percussion on top of the melody.

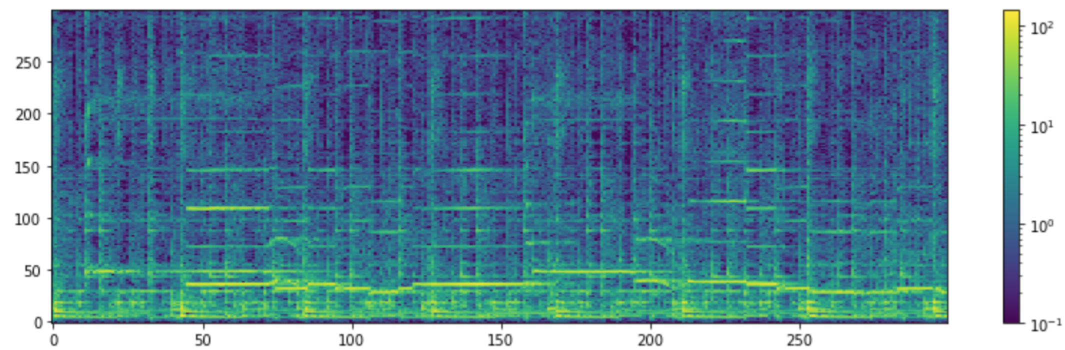

Fig. 5.3.1, spectrogram of Der_Voghormia_2.wav

Notice the mix of horizontal and vertical lines in the plot. The horizontal lines represent steady sinusoids with a narrow-band frequency range (harmonic), while the vertical lines represent impulses with a wide-band frequency range (percussion). Knowing this, we can split up a given audio signal into a sum of a harmonic signal and percussive signal. This technique is known as **harmonic-percussive separation (HPS)** and makes use of the iSTFT (Müller, 2015, p. 419). HPS comes in six steps:

1. Convert our time-domain audio signal to its STFT
2. Apply a harmonic filter to the STFT to create a harmonic-filtered STFT
3. Use the harmonic-filtered STFT to create a harmonic mask
4. Apply the mask to the original STFT to create a harmonic STFT
5. Repeat steps 2-4 with a percussive filter
6. Apply the iSTFT to the harmonic and percussive STFTs to recover our time domain harmonic and percussive signals. The original time-domain signal is equal to the sum of these two new signals!

For our choice of filtering, we make use of **median filtering**. Median filtering works by constructing a new signal whose contents are generated by taking medians in equally-sized windows of our original signal. This eliminates spikes in our signal, while retaining its edges. The figure shown displays the smoothing of a random signal using a window of size 5:

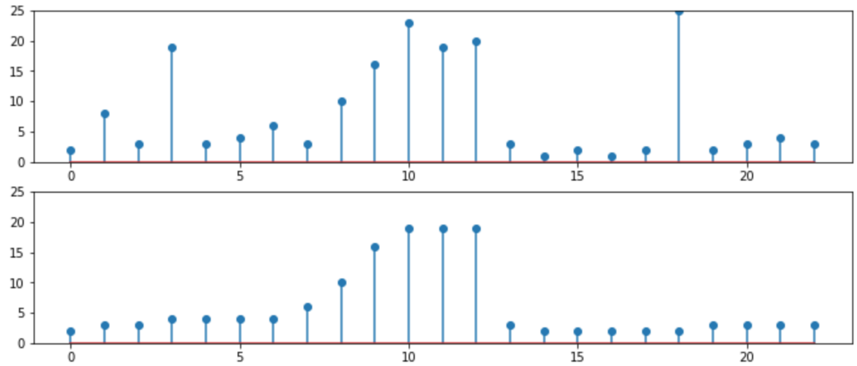

Fig. 5.3.2, Median filtering with window size 5

For our edge cases, we can zero-pad the original signal, meaning we extend the signal in a direction and give the new signal parts a value of 0. This is so our filtering window has enough of a signal to capture. Filtering the rows and columns to the original STFT enables us to create **masks**, which essentially assign cells of the STFT to be harmonic or percussive. These masks make use of Wiener filtering, and are determined by:

$$\mathcal{M}_h[n,k] = \frac{\tilde{\mathcal{X}}_h[n,k] + \frac{\epsilon}{2}}{\tilde{\mathcal{X}}_h[n,k] + \tilde{\mathcal{X}}_p[n,k] + \epsilon}$$
$$\mathcal{M}_p[n,k] = \frac{\tilde{\mathcal{X}}_p[n,k] + \frac{\epsilon}{2}}{\tilde{\mathcal{X}}_h[n,k] + \tilde{\mathcal{X}}_p[n,k] + \epsilon} \qquad \text{(Eq. 5.3.1)}$$

Where the tildes represent our filtered spectrograms, *h*/*p* denote harmonic/percussive, and ϵ is a very small constant of our choice (typically in the range 10^{-2} - 10^{-5}). We then multiply these masks by the original STFT to create our harmonic and percussive STFTs. Finally, applying the iSTFT to our two new STFTs returns our original signal, but now separated into its harmonic and percussive components. Listening to these separated signals will support this idea, but it is possible (and likely) for some artifacts to remain after separation.

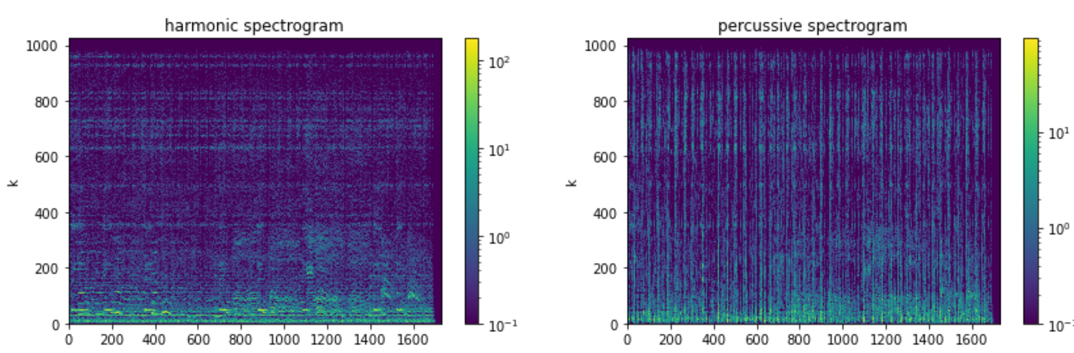

Fig. 5.3.3, HPS of Der_Voghormia_2.wav with ϵ = 0.0001

5.4 Chord Recognition and Scoring

Now that we can reliably extract a harmonic region of interest, we can computationally compare the chord structure of one song versus another. We once again look at the chromagram, and process a chord template over it. We will use a **major/minor triad** chord template, but the process can be extended to augmented/diminished triads, seventh chords, and others. This template is a matrix of triads “in order”, meaning the major triads are listed first (beginning with C major = 0) and are followed by the minor triads:

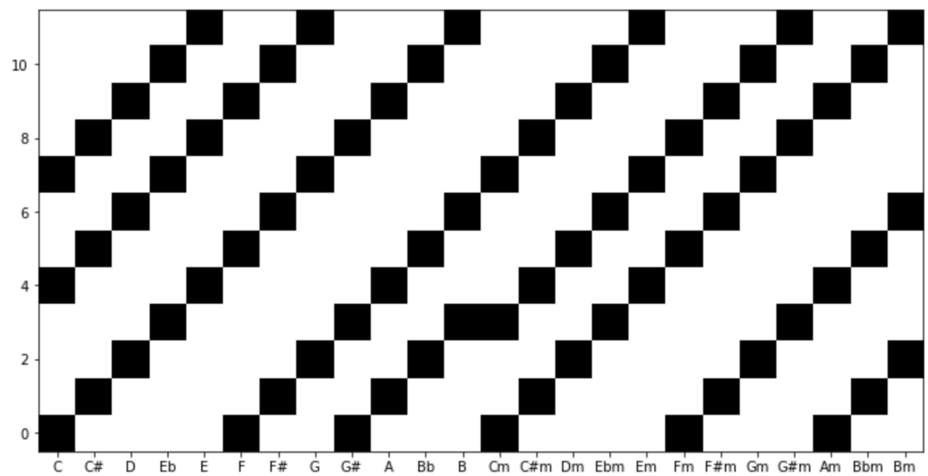

Fig. 5.4.1, Major/minor triad chord template

Multiplication of this against the chromagram of a song will result in a chromagram-like score matrix, where each row is the score a particular chord received. The chord with the highest score is the algorithm’s best estimate for what chord occurred at that time. For Der_Voghormia_1.wav, this score matrix looks like:

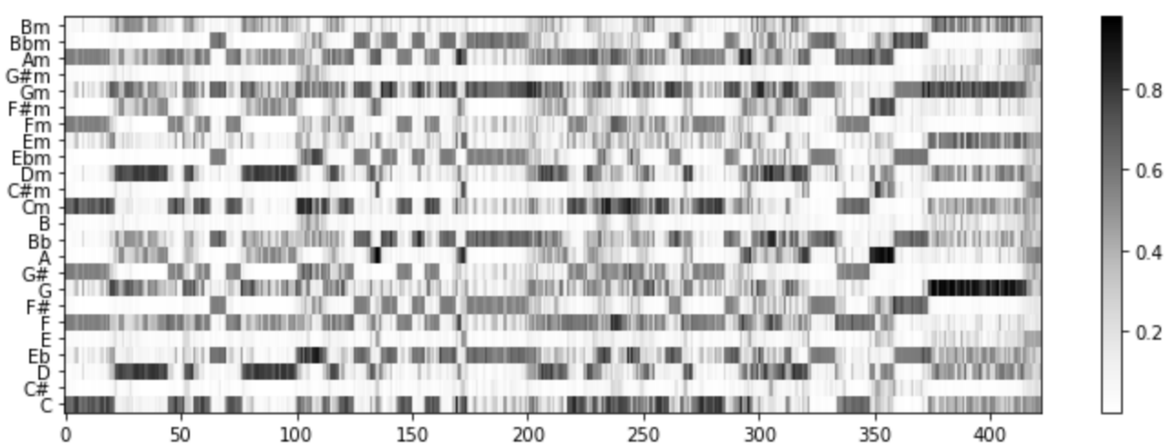

Fig. 5.4.2, Der_Voghormia_1.wav score matrix

The score is determined by the normalized dot product between the chroma vector **x** and the chord template **c**, which is given by:

$$\mathrm{Score}(x, c) = \frac{\langle x,c\rangle}{||x||\,||c||} \qquad \text{(Eq. 5.4.1)}$$

As depicted in (Fig. 5.4.2) above, a darker shading means a higher confidence in a specific chord. One thing to note is the influence of overtones, which can impact our scoring and identification. One way to handle this is to introduce an exponential decay factor α to simulate overtones over the ideal template. This adjustment modifies our template by giving the harmonics a decline in energy rather than forcing them to zero:

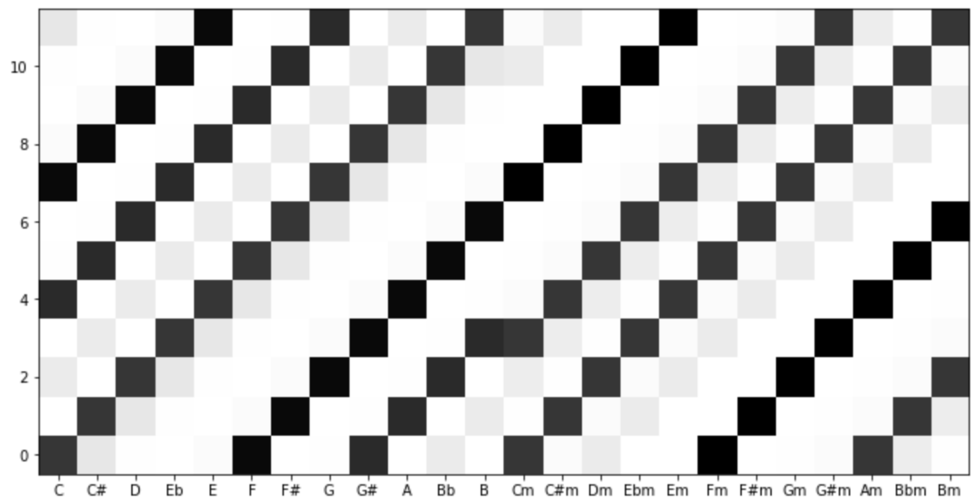

Fig. 5.4.3, Modified chord template with α = 0.5

Furthermore, the tuning between two pieces may not be the same. A4 = 440 Hz is not a requirement, and how many cents the tuning differs by can drastically alter the chord recognition and scoring. This can be solved by passing a tuning parameter into the creation of chromagrams. Knowing the tuning difference is extremely difficult if not impossible by ear, so an algorithm was written to discover the best tuning. This algorithm simply processes the score calculation using tuning values in a specified range, and outputs the tuning with the best score. For our purposes, a step size of one cent ranging from -50 cents to +50 cents (half a semitone each way) works best. This captures the most reasonable range of tuning differences while maintaining a respectable computation speed.

Despite all this, further improvements still exist and will be explored. In the next section, we will discuss two additional filtering approaches to further improve the quality of our chord recognition.

Chapter 6: Design Improvements

In this chapter, we will look at two different filtering methods to improve our chromagrams and our chord recognition.

6.1 Pre-filtering

As discussed in chapter 5.1, spectral smoothing is only one way to enhance a chromagram. We can also apply **temporal smoothing** to the chromagram. This is known as a **pre-filtering** method, and is used to avoid irrelevant local variations (Müller, 2015, p. 271). Pre-filtering smooths out any temporal fluctuations in the chromagram, making the overall structure more consistent. To perform temporal smoothing, we apply an averaging filter on each row of the chromagram. This filter takes a window of size ***L*** centered at our choice of ***n***, and averages the local values:

$$x_L[n] = \frac{1}{L} \sum_{l=-(L-1)/2}^{(L-1)/2} x[n+l] \qquad \text{(Eq. 6.1.1)}$$

This is a simple process to clean up some of the time-axis mess a chromagram could have when notes rapidly change over a short period of time, and plays a part in generating a CENS chromagram as discussed in chapter 5.1.

Instead of performing pre-filtering, we can also approach chord recognition by means of **hidden Markov models (HMMs)**, which is a **post-filtering** method that will be discussed in the next section (Müller, 2015, p. 291). Using HMMs makes pre-filtering redundant, as it contains context-aware smoothing within the process. As such, it will be of interest to score a pre-filtered chromagram against a post-filtered chromagram and see which method performs better.

6.2 Hidden Markov Models

HMMs build off of the idea of a **Markov chain**. A Markov chain represents a set of discrete states, and a set of the probabilities of transitioning between these states. A state can transition to itself, and the sum of the transition probabilities must equal 1. In our musical context, let's say we only have three chords in a song we wish to make: C major, F major, and G major. An example Markov chain could be modeled like:

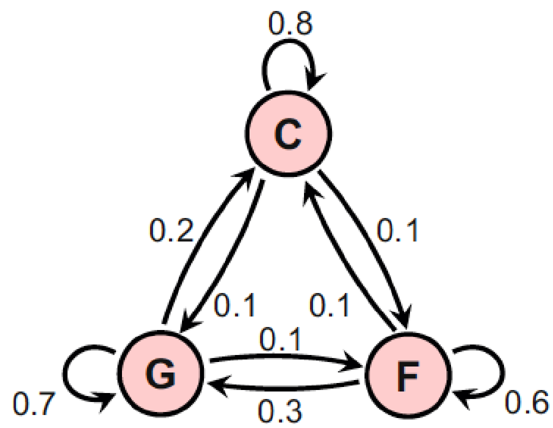

Fig. 6.2.1, Markov chain of three major triads (Müller, 2015, p. 275)

We can refer to an initial probability vector to determine our initial state, then run the process repeatedly until we have enough chords for the song we want to make. We can expand on the idea of a Markov chain to construct a model that can predict where a song wants to move (or stay).

A *hidden* Markov model is a more statistical expansion of the Markov chain. Our chords are our states just like before, but we now have an observation layer added to our model. In this case, the observation layer is our chroma vectors. Every time a state is active, we have an observation produced. However, we cannot directly observe the underlying chord, only the chroma vectors that are mapped to them. This gives the model its name of "hidden".

There is a problem with our chroma vector observations - their compositions are twelve dimensional. This is a computational mess, and describes much more different chroma vectors than we will care about. To avoid this, we can discretize our model by first creating a codebook of finite observations. We then quantize the chroma vectors by generating a finite mapping between the codebook and our actual observations. Luckily for us, we have a process that can already do this. We can create a size 24 codebook for our 24 major and minor triads, and we can map our observations to them by finding the best match through our same scoring process described in chapter 5.4.

There are five distinct parameters that define an HMM. Similar to a Markov chain, we have our set of discrete states, a matrix of transition probabilities (which we will shorthand as the *transition matrix*), and a vector of initial state probabilities. Some chord transitions are more likely than others, but it is difficult to quantify this across every piece of music ever performed or composed. Therefore, a uniform approach to the transition and initial state matrices is a safe approach, albeit lacking in musical sophistication. Specifically for the transition matrix, we can have an emphasis along its main diagonal to reinforce the idea of a chord transitioning to itself, but otherwise has an equal chance to transition to any different chord. This emphasis is simply a weighting factor on the diagonal that makes it stand out more compared to the other values in the matrix.

In addition to the aforementioned three parameters, we must have observations, and must generate a matrix of output probabilities known as the *emission matrix*. The observations have been discussed to be the 24 different major and minor triads, so that leaves the generation of the emission matrix.

For a given state, the probability distribution amongst the different observations isn't necessarily uniform. For example, C major chords can produce observations that look like A minor or E minor due to their composition. Therefore, we will want to assign probability values based on how similar each chord template is to another. We reintroduce the decay factor for the overtones into this assignment, and end up generating a template is similar to the ideas presented in Fig. 5.4.3:

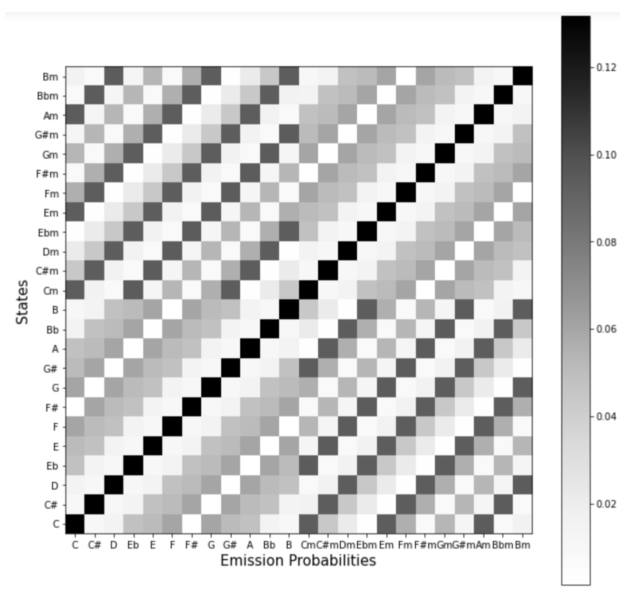

Fig. 6.2.2, Emission matrix with α = 0.5

We now have our five parameters, and thus have a complete HMM. To process our model, we can run the **Viterbi algorithm** on it. The Viterbi algorithm recursively computes optimal state sequences by making use of a backtracking procedure. This is a dynamic programming algorithm that is quite similar to DTW as presented in chapter 5.2. What it accomplishes is that given an observation sequence, what is the most likely sequence (or, what is the highest-scoring sequence) out of all the possible sequences from our observations (Müller, 2015, p. 281). The outcome of this algorithm is influenced by our transition matrix's diagonal weighting, as well as our choice of decay factor α.

We now have several different optimizations to compute a score from. The next section will detail how these methods fare, with some optimizations being combined with others:

- Spectral smoothing
- Overtone adjustments
- Tuning adjustments
- Pre-filtering
- Post-filtering (HMMs)

Chapter 7: Results

Results will be given through the following subsections, which will display the spectrograms and log-compressed chromagrams of each song side by side, followed by their DTW, and then the HPS of the comparing song. At the end, a table of scores will be presented along with a discussion regarding their values. As discussed in chapter 5.4, the score is determined by the normalized dot product between the chroma vector of our comparing song, and the chord template generated from Der_Voghormia_1, our reference song.

7.1 Parameters

Parameters to keep in mind are the logarithmic compression factor γ for chromagram enhancement, Wiener filtering constant ϵ for our HPS masks, decay factor α for the chord template overtones, tuning adjustment ***t***, prefiltering window size ***L***, and diagonal weighting factor ***w*** for our transition matrix. We will look at α***, t, L,*** and ***w.*** We will fix γ = 1 and ϵ = 10^{-4}, but note that improvements can be made through adjusting these two parameters.

In addition, the Fourier transform parameters used were a sampling rate F_s = 22050 Hz, DFT window length ***N*** = 4096, and hop size ***H*** = 2048. Powers of 2 work best ***N*** and ***H*** due to the design of the DFT and STFT, and found that ***H*** = ***N***/2 yielded favorable results. These parameters can be played with, but their overall impact on the chord recognition is not as significant or as interesting as the other parameters.

7.2 DV1 and DV2

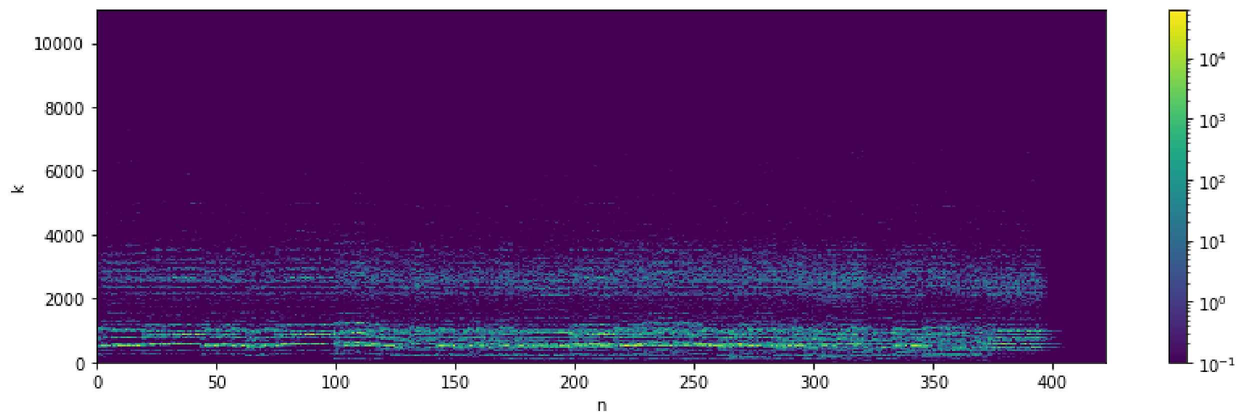

Fig. 7.2.1, Spectrogram of DV1

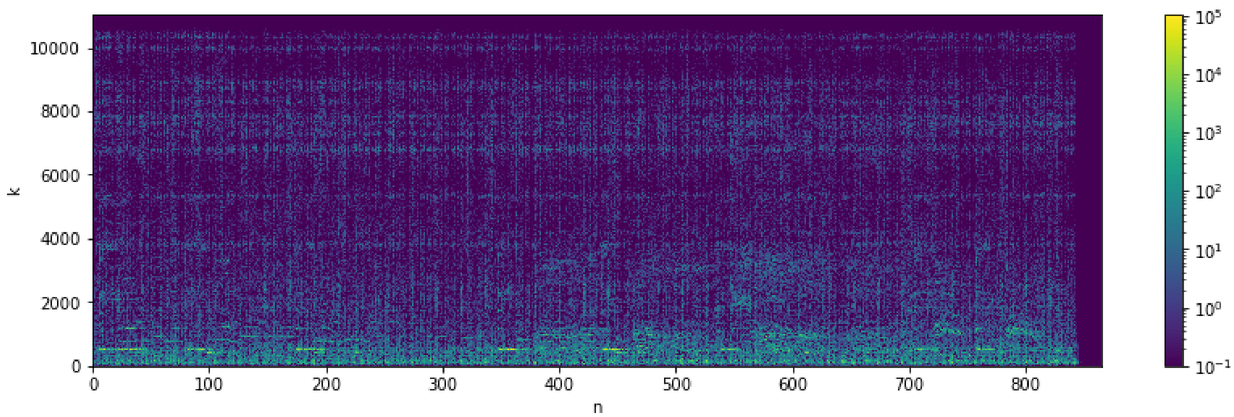

Fig. 7.2.2, Spectrogram of DV2

DV1 is digitally generated audio that was transcribed from a physical manuscript, and features three tenor voices and one bass voice.

DV2 is a professional studio recording that features percussion and duduk (a woodwind instrument) in addition to its vocals.

One main observation here is the differences in the y-axis. There is a much wider range of frequencies being presented in DV2, but the main concentration is around the lower values. DV1 also has a notable gap between its two different colored bands, whereas in DV2 there is substance along most rows.

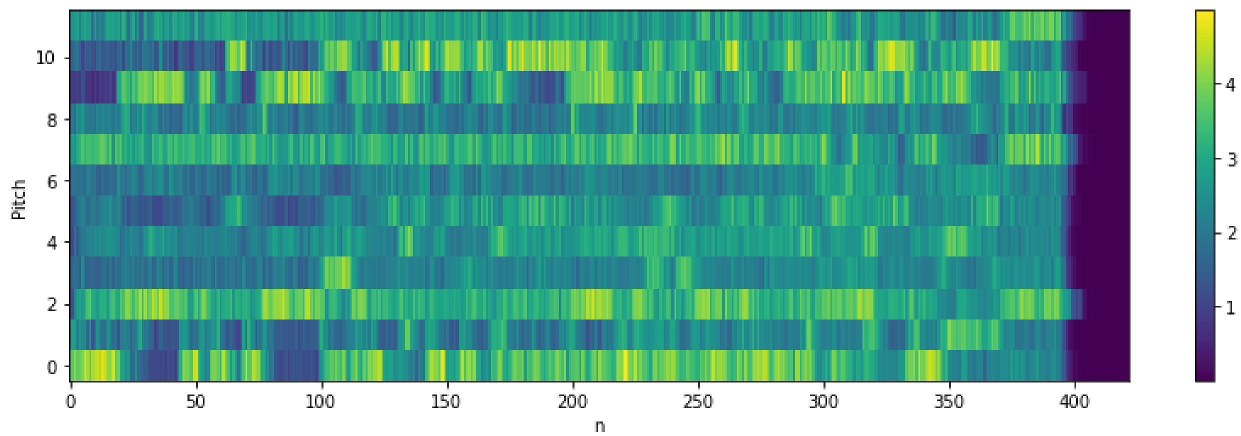

Fig. 7.2.3, Chromagram of DV1

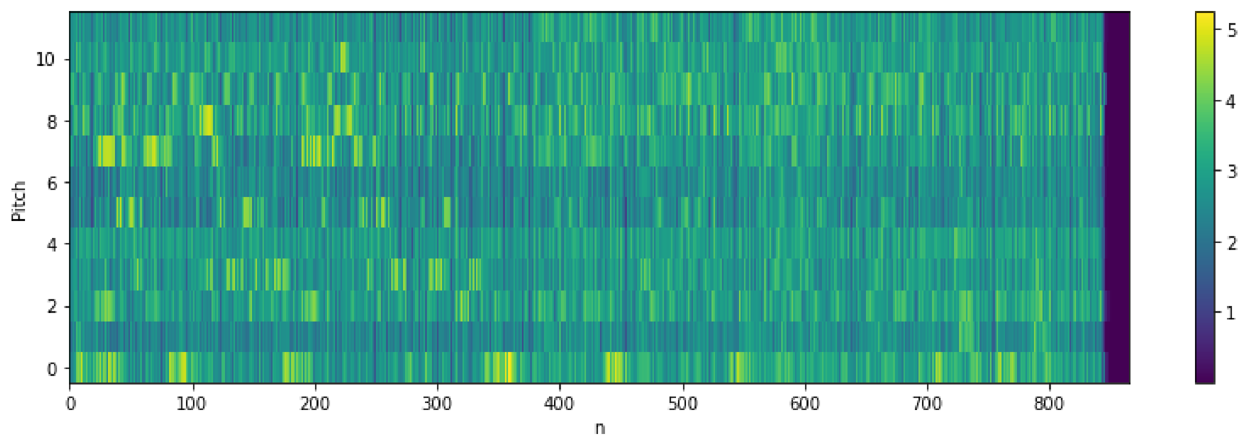

Fig. 7.2.4, Chromagram of DV2

Here we see our differences in notes. Although there seems to be an emphasis around 0 = C and 7 = G, DV1 has higher intensities of 10 = B-flat than DV2 does. The near constant presence of this B-flat in DV1 will influence our chord identification by trying to fit the many C's and B-flat's into chords, whereas DV2 has more of a note spread that can lead to the detection of different chords.

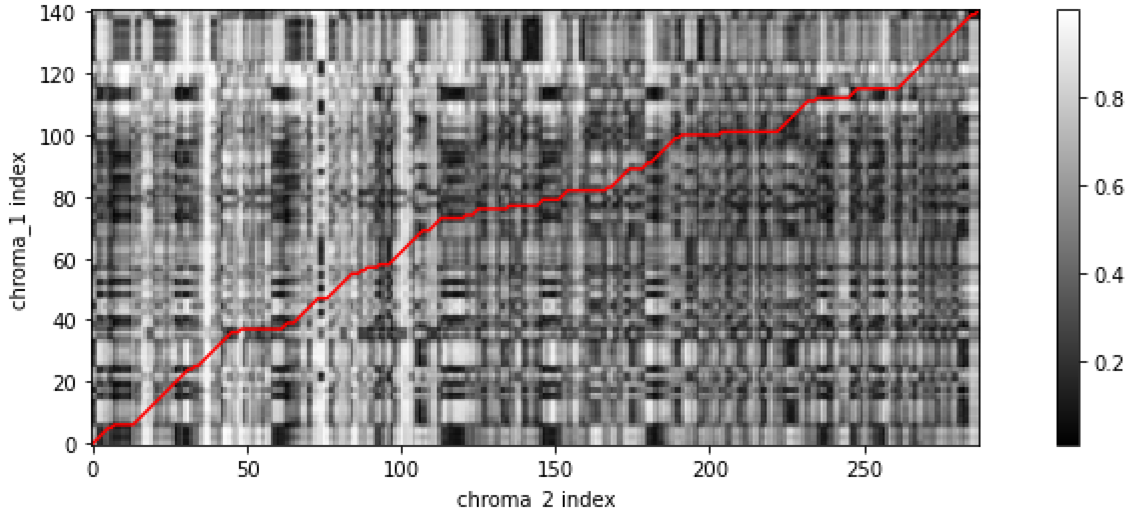

Fig. 7.2.5, DTW between DV1 and DV2 (CENS)

From this DTW, we see that there is a pretty consistent matching of melody due to the path mainly being a diagonal from one corner to the other. Listening to DV2 supports this idea, as the song is a repetition of the DV1 melody but in different contexts (duduk performing the melody, voices performing the melody). The bumps along the way can be explained by breaks between the melodies, as well as the underlying percussion.

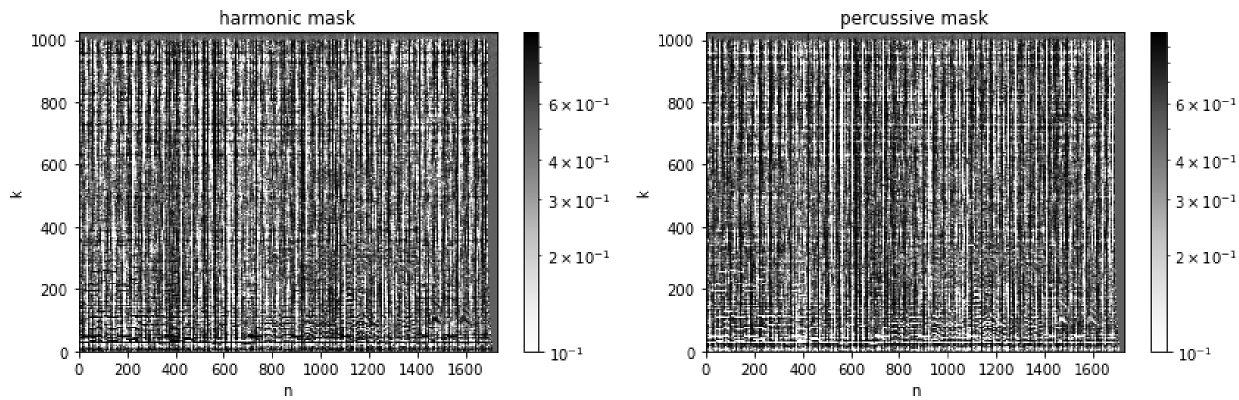

Fig. 7.2.6, HPS masks for DV2

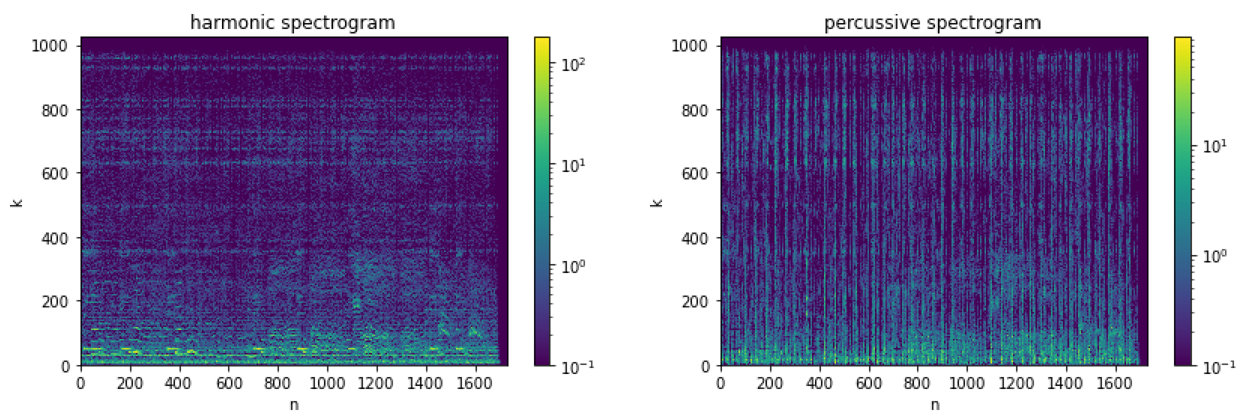

Fig. 7.2.7, Separated spectrograms for DV2

The masks and separations confirm that DV2 features a good amount of percussion, which has been removed from the harmonic audio. Listening to this separation confirms that it mostly worked, but some percussion remained in the harmonic signal. Despite this, the percussion sounds were much lower and the harmonic content was much clearer.

7.3 DV1 and DV3

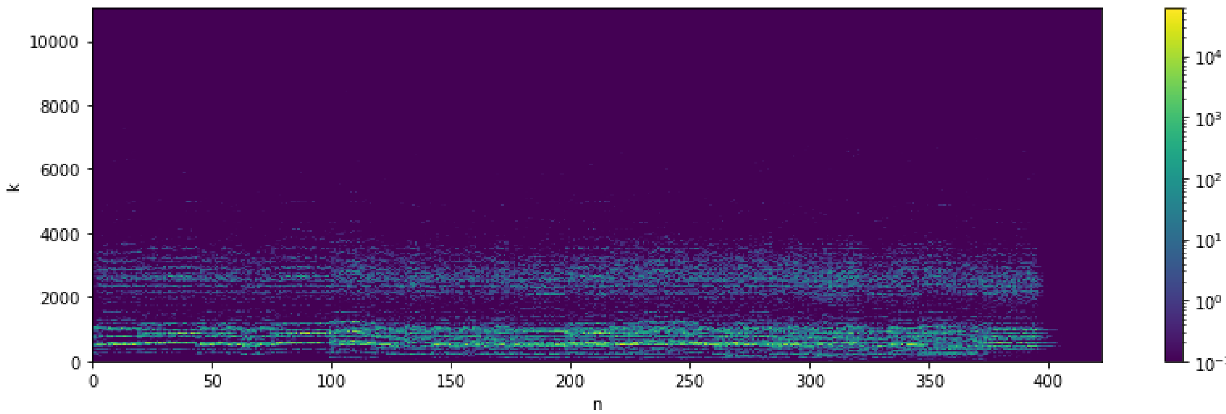

Fig. 7.3.1, Spectrogram of DV1

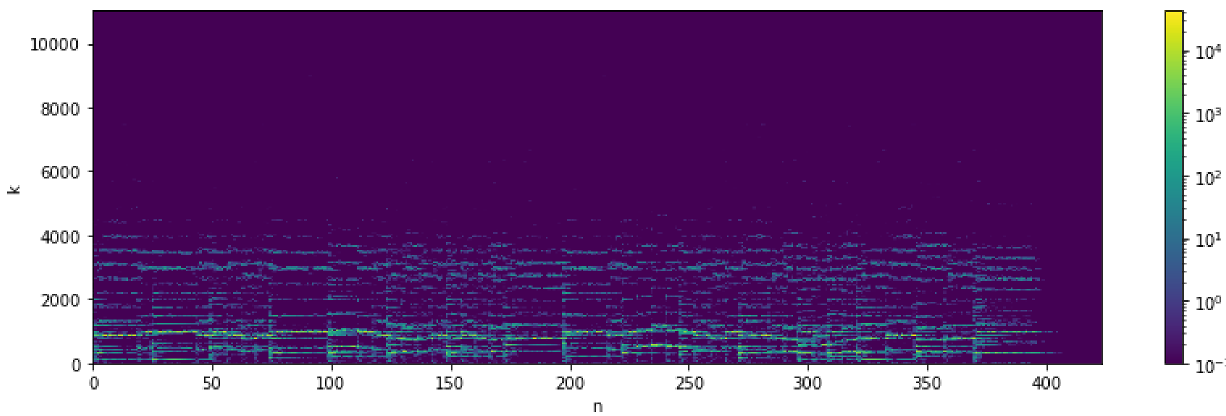

Fig. 7.3.2, Spectrogram of DV3

DV1 is digitally generated audio that was transcribed from a physical manuscript, and features three tenor voices and one bass voice.

DV3 is digitally generated audio that was transcribed from a physical manuscript, and features one solo voice and piano accompaniment.

The spectrograms make it clear how similar these songs are on a visual level - the frequencies present are similar, and the number of samples in each song are almost exact. Despite this, the harmonic structure will vary due to the chords present in the piano accompaniment being different from the chords created by the supporting voices in DV1. In addition, the spacing between rows in DV3's spectrogram is more pronounced than the rows in DV1's spectrogram.

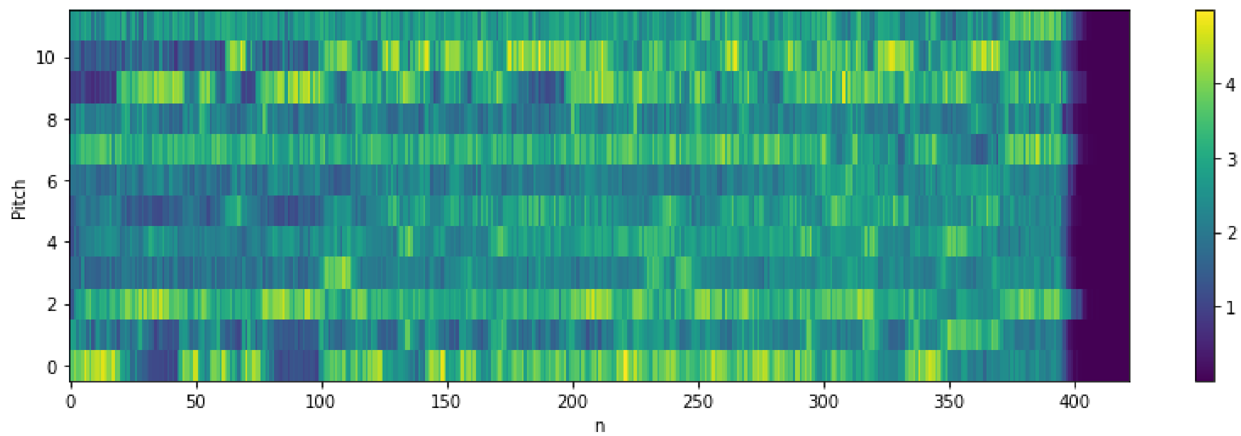

Fig. 7.3.3, Chromagram of DV1

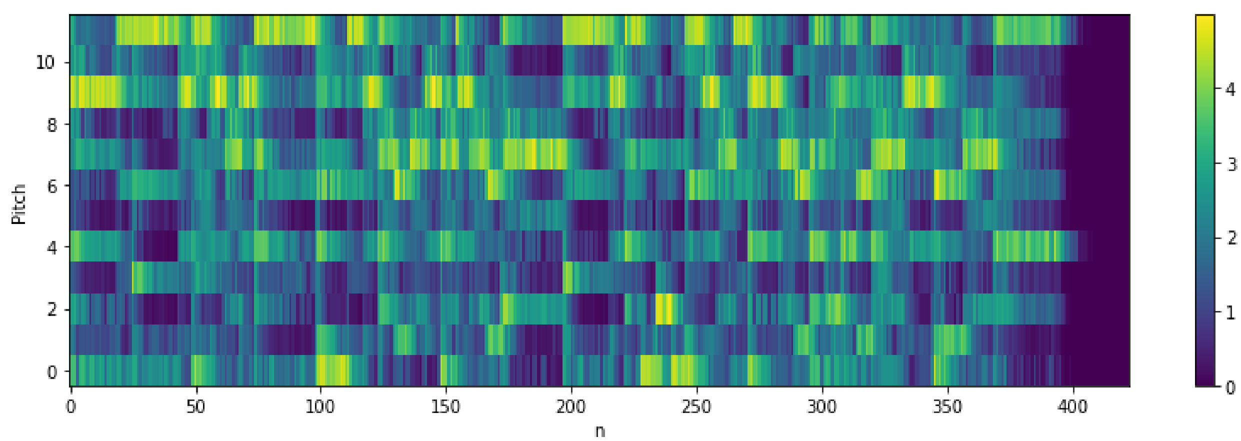

Fig. 7.3.4, Chromagram of DV3

The chromagrams here really show how these two songs mainly differ by three half-steps. DV1 starts on 0 = C, whereas DV3 starts on 9 = A. The intensity patterns throughout the chromagrams are very similar, which can be seen with 9 = A and 11 = B in DV3's chromagram having the same pattern as 0 = C and 2 = D in DV1's chromagram.

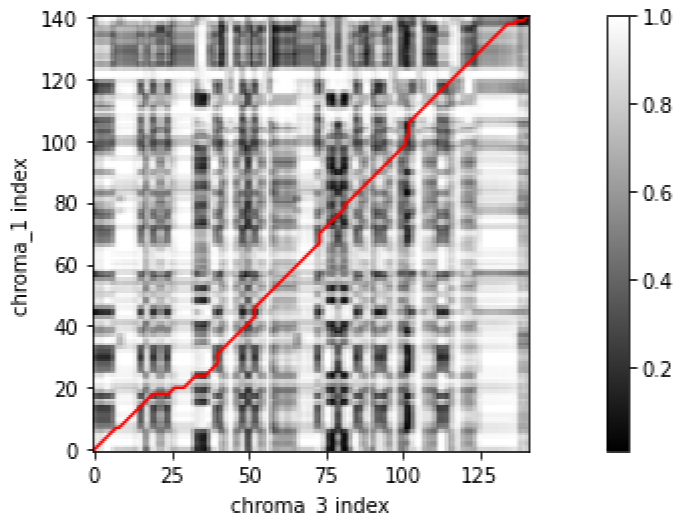

Fig. 7.3.5, DTW between DV1 and DV3 (CENS)

The square shape and the persistent diagonal really hammer in that the two songs are melodically identical. The minor bumps are due to the differences in the underlying harmony, and how they could cloud the main harmony from the algorithm.

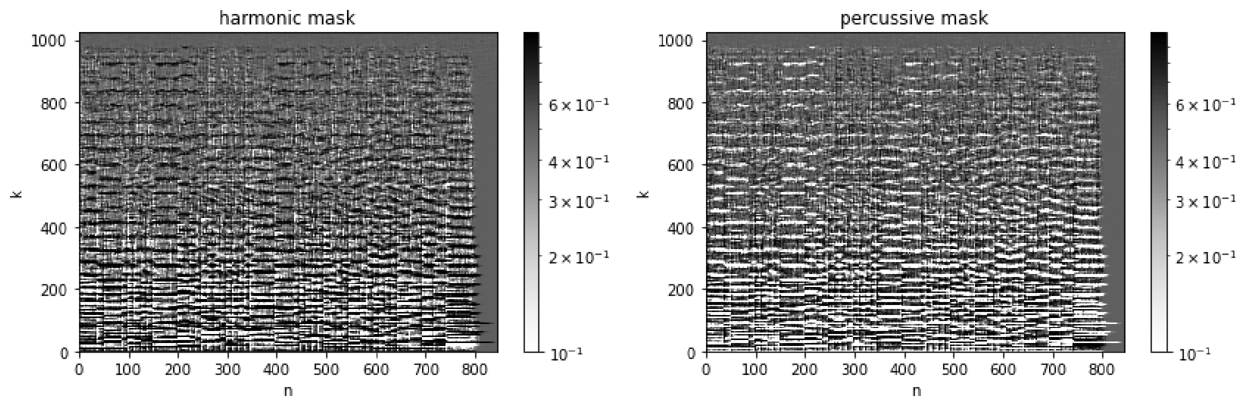

Fig. 7.3.6, HPS masks for DV3

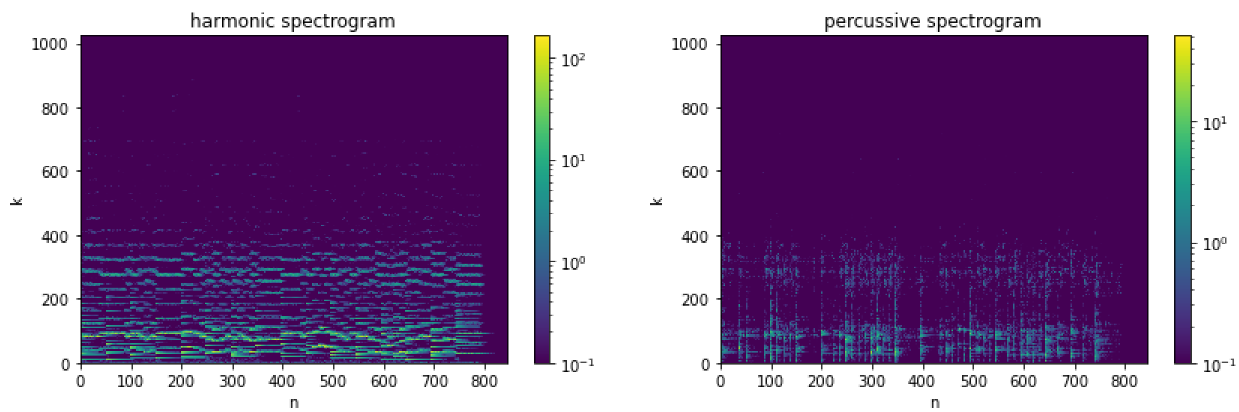

Fig. 7.3.7, Separated spectrograms for DV3

Even though HPS was not necessary for this song, it may be surprising to see the percussion spectrogram have some content. This can be due to the masks and filters expecting percussive content in the first place. The mask design makes use of the non-zero constant ϵ, meaning that even with no percussive content, the filters will find something to extract. The percussive signal plays a cacophonous version of the original song, with the voice and piano parts extremely distorted. It is difficult to aurally tell the difference between the original song and the harmonic signal, but there technically are some harmonic components accidentally extracted.

7.4 DV1 and DV4

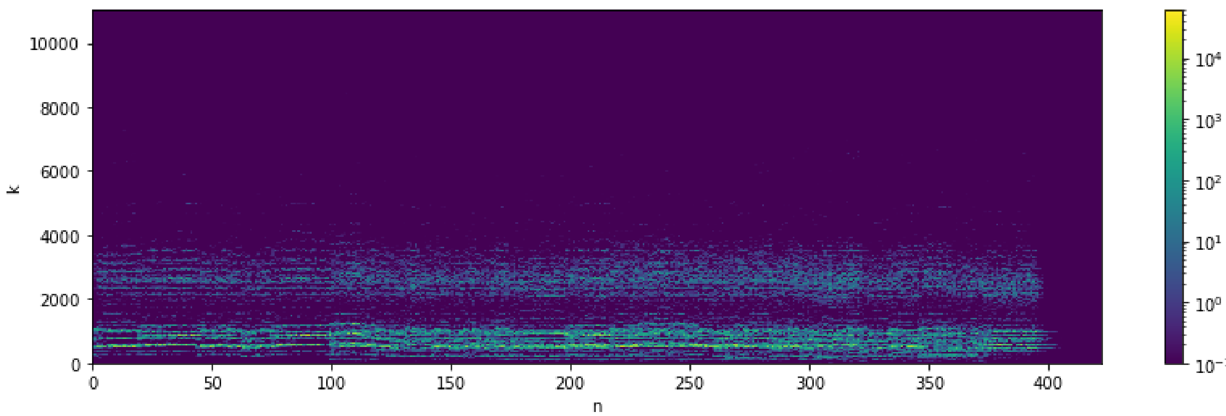

Fig. 7.4.1, Spectrogram of DV1

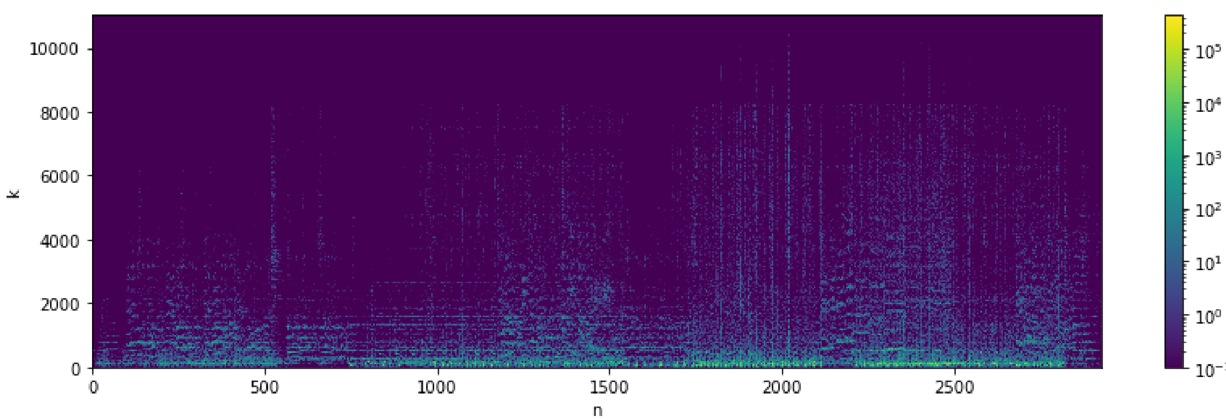

Fig. 7.4.2, Spectrogram of DV4

DV1 is digitally generated audio that was transcribed from a physical manuscript, and features three tenor voices and one bass voice.

DV4 is a live band performance featuring voice, drums, strings, and electric guitar that incorporates a funk groove to support the main melody.

Once again, we see most of the content condensed in the lower area of the spectrogram. Notice how there are fading peaks extending upwards. These can be the product of the different instrumentation, as well as noise since we are working with a live recording that is unfiltered.

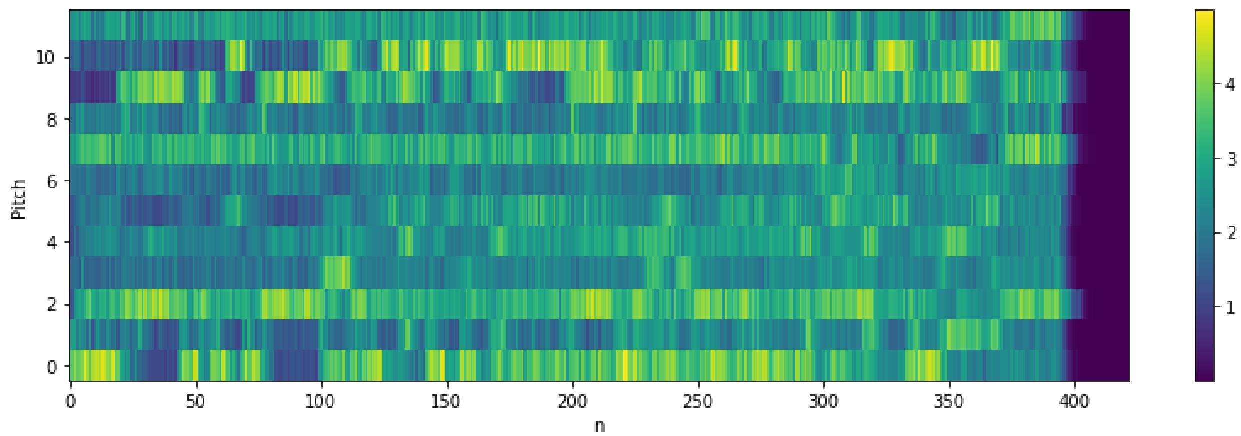

Fig. 7.4.3, Chromagram of DV1

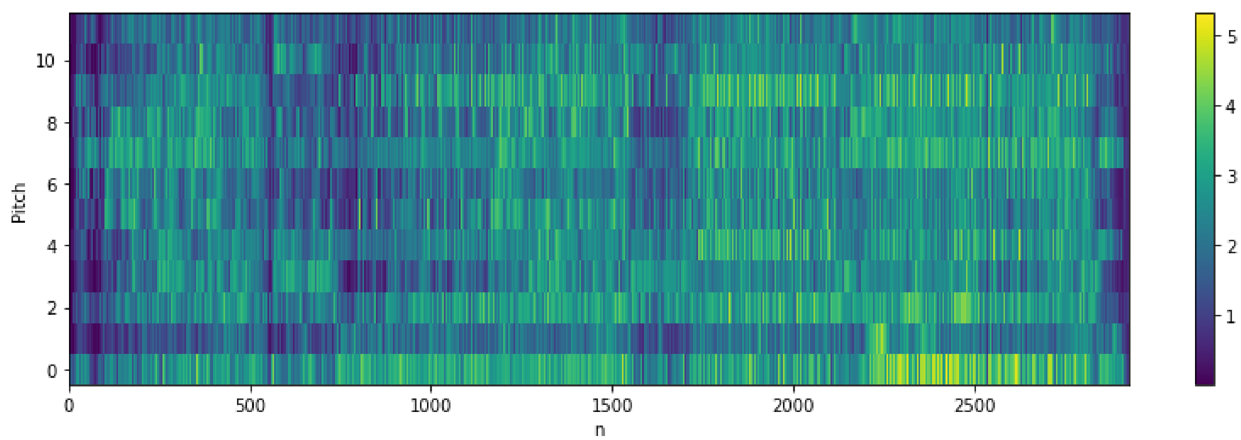

Fig. 7.4.4, Chromagram of DV4

There is not much to compare with these chromagrams. They suggest that the songs are pretty different, as the intensities don't line up in similar places. Furthermore, DV4 lacks intensities earlier on which makes it harder to tell what is going on melodically.

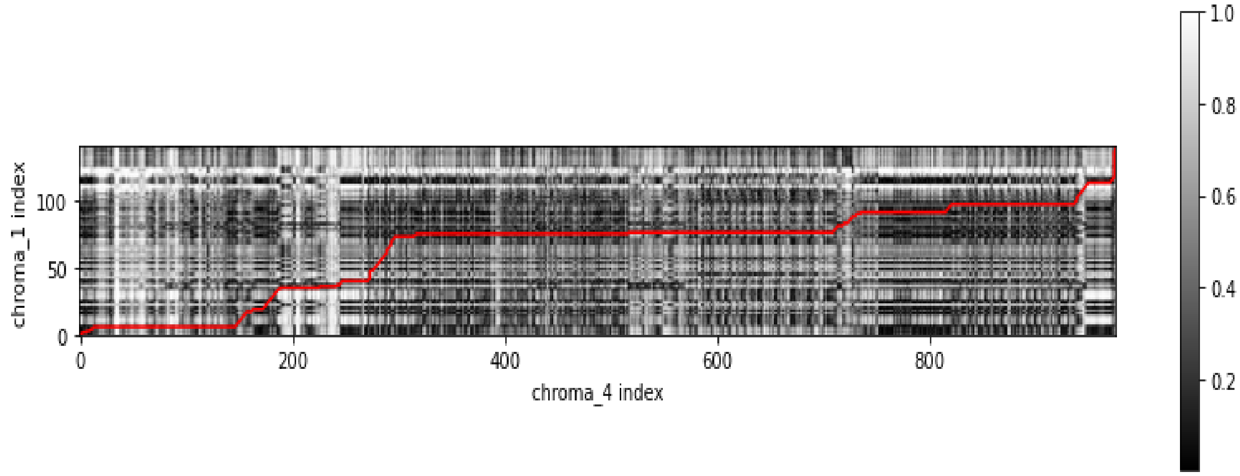

Fig. 7.4.5, DTW between DV1 and DV4 (CENS)

DTW confirms our earlier suspicion that the two songs are considerably lacking in similarities, due to the many horizontal lines present. There are some diagonal-like areas especially earlier on, but overall the region matching is not very good. Rather than taking a section, the whole song will be compared against DV1.

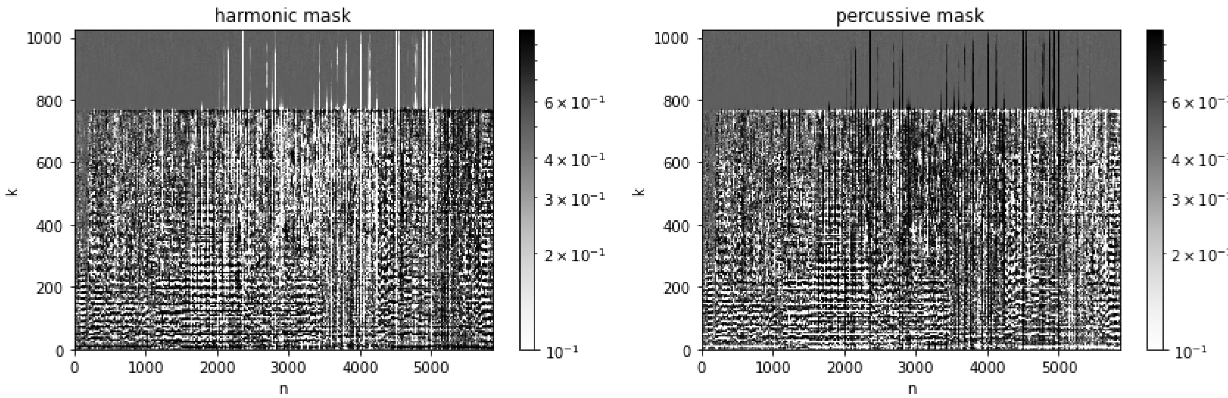

Fig. 7.4.6, HPS masks for DV4

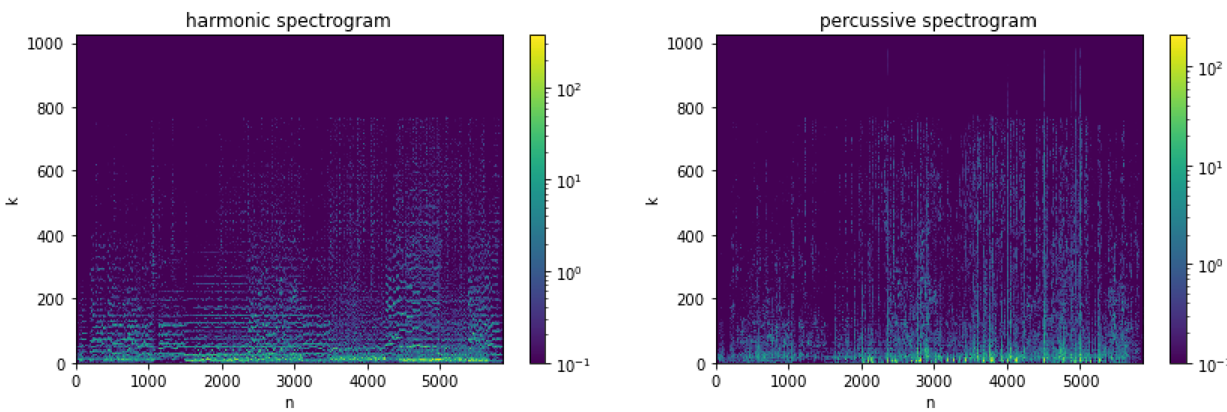

Fig. 7.4.7, Separated spectrograms for DV4

The separation looks pretty good. We know that there was percussive content, and it appears that a lot of it was filtered out from the harmonic components. However, listening to the separated signals tells a different story. The percussive signal sounds like it did a good job, but the harmonic signal is missing a lot of itself. The audio sounds “static-y” or that it “crackles”, and the harmonic content is made very unclear. Our HPS algorithm doesn’t do a great job with DV4.

7.5 DV1 and DV5

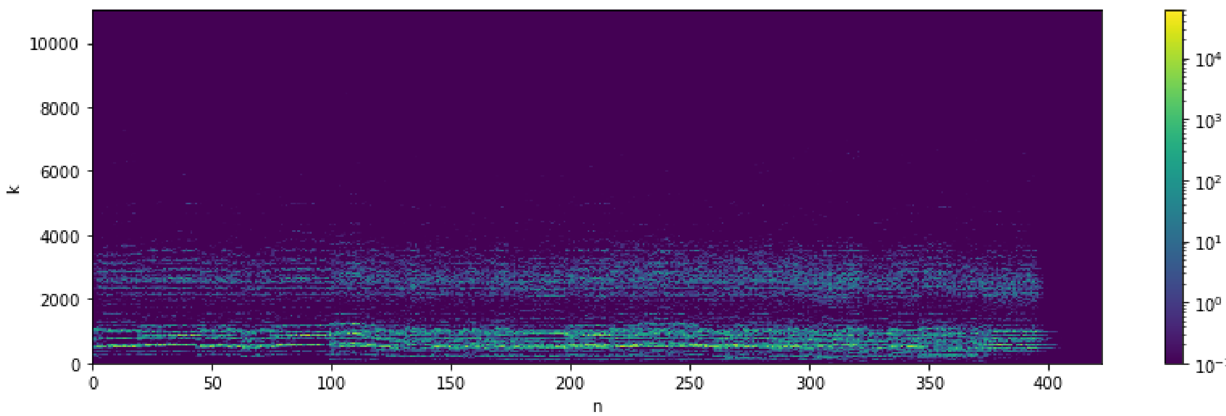

Fig. 7.5.1, Spectrogram of DV1

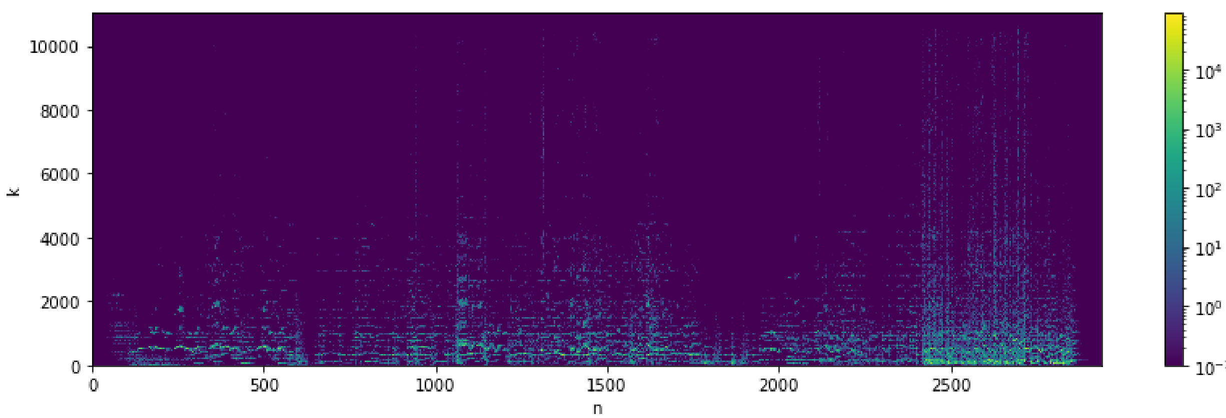

Fig. 7.5.2, Spectrogram of DV5

DV1 is digitally generated audio that was transcribed from a physical manuscript, and features three tenor voices and one bass voice.

DV5 is a professional studio recording that features a medley arrangement of 3 different variations.

A lot of DV5's content is towards the end of the piece where the third variation is. Otherwise, for the most part, it shares a similar sparisity early on with DV1. DV5's spectrogram does lack the two horizontal bands of spectral content that is pronounced in DV1's spectrogram.

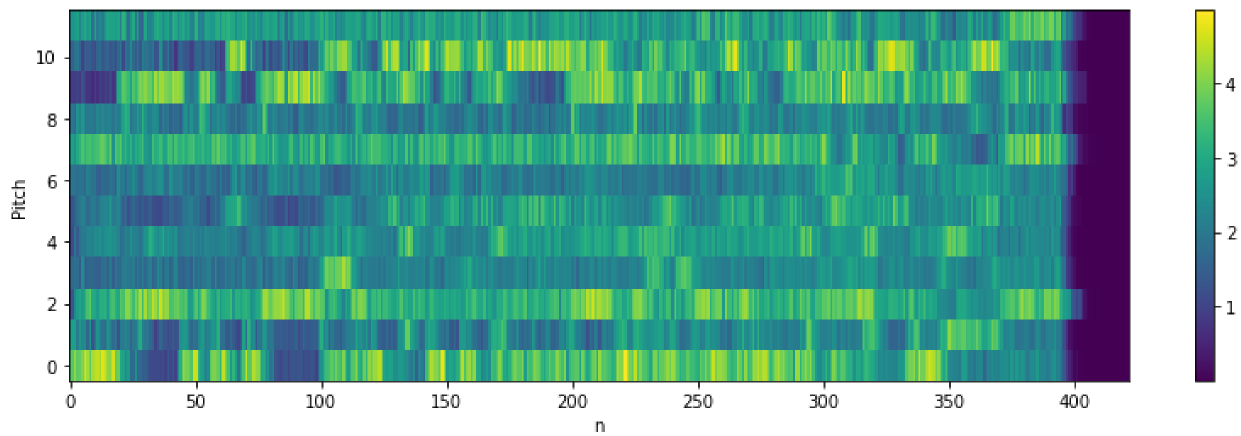

Fig. 7.5.3, Chromagram of DV1

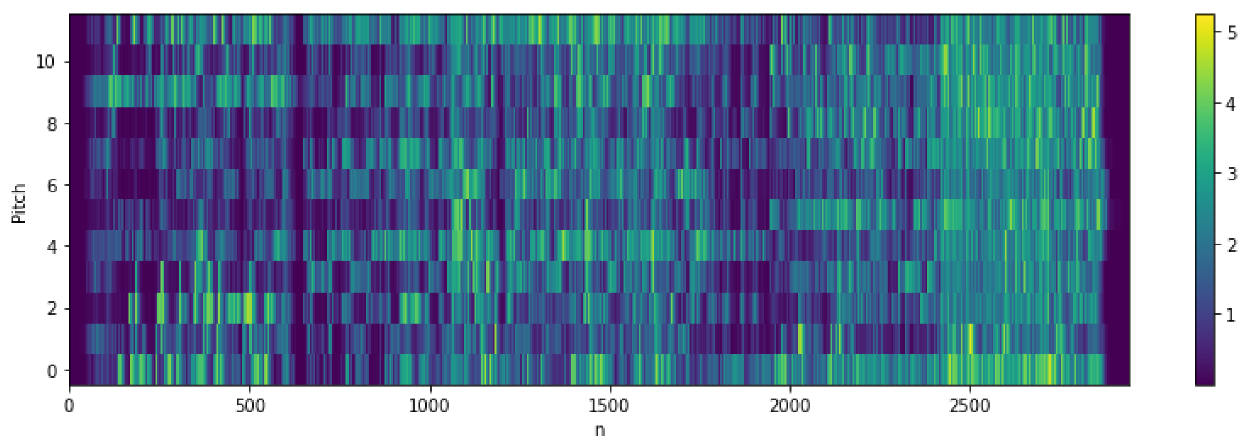

Fig. 7.5.4, Chromagram of DV5

Similarly to the spectrogram discussion, the content of DV5 is backended. There are some intensities in similar places between DV1 and the end of DV5, suggesting that the chromagram for the third variation of DV5 looks like a condensed version of DV1's chromagram.

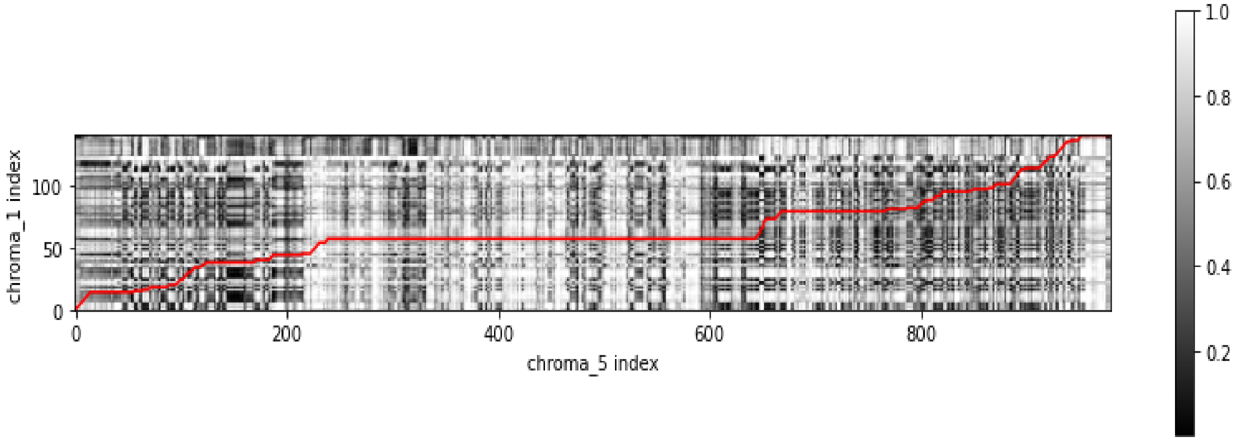

Fig. 7.5.5, DTW between DV1 and DV5 (CENS)

Unlike DV4, the large horizontal line isn't a problem. DTW detects two decently matching regions at the beginning and the end - the first and third variations. The third was deemed the best fit to score against DV1 based on the spectrograms, chromagrams, and DTW. In addition, an aural comparison makes it clear that the rhythm between DV1 and DV5 is most similar in the third variation. A more similar rhythm means less chord detection errors in general by minimizing the differences (fluctuations) between pieces.

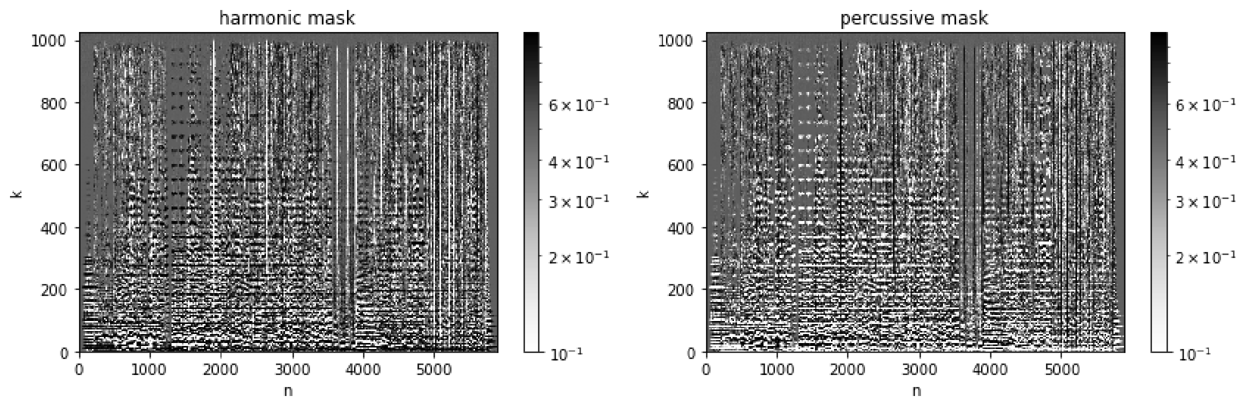

Fig. 7.5.6, HPS masks for DV5

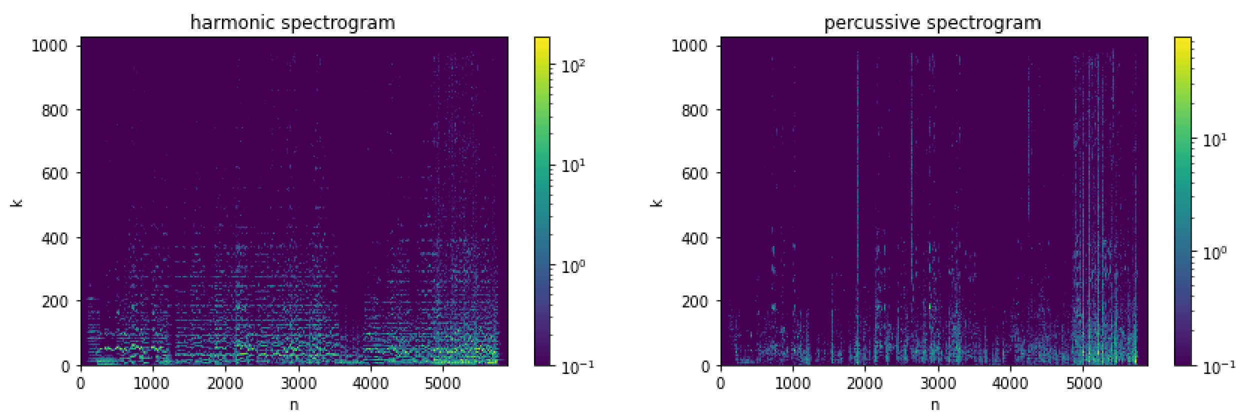

Fig. 7.5.7, Separated spectrograms for DV5

As mentioned before, the third variation features percussion, which is supported by the separated spectrograms - the percussive spectrogram has a lot of content towards the end of the piece. The separation worked pretty well for this piece, but was not perfect. Although the percussion was successfully filtered out, the harmonic quality took a hit. Unlike DV4, the melody and harmony was still recognizable.

7.6 Table of Scores

Scores are calculated against the chords from DV1 based on (Eq. 5.4.1), taking on values from 0 to 1. A score close to 0 means that the two songs, despite sharing the same melody, are harmonized in entirely different ways. A score close to 1 means that the vast majority of the chords between the two songs matched. Note that the prefiltered and HMM scores used the compression and tuning optimizations. The decay factor α for the chord template overtones, tuning adjustment ***t***, prefiltering window size ***L***, and diagonal weighting factor ***w*** for our transition matrix are also shown.

Song	Unoptimized Score and α	Compressed + Tuned Score and ***t***	Prefiltered Score and ***L***	HMM Score and ***w***
DV2	0.15242 α = 0.2	0.18374 ***t*** = -10 cents	0.27455 ***L*** = 13	0.29114 ***w*** = 9
DV3	0.13081 α = 0.3	0.29147 ***t*** = -298 cents	0.29147 ***L*** = 1	0.34621 ***w*** = 2
DV4	0.08427 α = 0.5	0.09322 ***t*** = +20 cents	0.17773 ***L*** = 99	0.14933 ***w*** = 3
DV5	0.10663 α = 0.7	0.10900 ***t*** = +1 cent	0.15121 ***L*** = 27	0.16744 ***w*** = 3

We can notice significant improvements in chord detection through our optimizations methods. Although no scores were able to be close to 1, the two filtering processes especially pose a higher quality comparison between songs, with HMM performing better than prefiltering most of the time.

DV2 suffers from the heavy amount of percussion and non-lyrical sounds that harmonize the melody. Even with HPS, some of this percussion remains in the harmonic extraction. Percussion can still provide harmonic content after all - the resonance of a drumhead can have a note associated with it. Improving the HPS algorithm could definitely help improve the score for this version of “Der Voghormia”.

Despite DV3 also being a digitally generated audio file, its score isn't as close as it should seem even with the tuning adjustment. Comparing the manuscripts, DV3 is three half-steps down from DV1, so one would think that a tuning adjustment should have it score close to 1. However, the main issue lies in the underlying harmony. DV1's melody is supported by two tenors and a bass, whereas DV3's melody is supported by a piano that plays different chords compared to DV1. Of course, some essence of similarity was still detected. Prefiltering is shown to have no impact on the score, which makes sense as digital audio is relatively cleaner than a live recording.

DV4 had a lot of noise in the audio due to the nature of the recording. This can explain why its scores were relatively low. Using a better recording or implementing a noise filter could see dramatic improvements for this song, especially in terms of improving its HPS and its scoring. The funk groove to this rendition also gives rise to harmonies that would not be present in DV1, messing with the chord detection. The recording quality could also explain how prefiltering had a greater impact on the score than HMM.

DV5 contains three unique imaginations of “Der Voghormia”. The rhythmic design can change how the algorithm detects a chord change, and one version also features percussion. The version with percussion was determined to be the most similar. Despite being the most similar version, the rendition still contains different ornamentations and accompaniment that impacts our chord detection.

Chapter 8: Conclusion

As just discussed, the scores are somewhat low. This isn't necessarily a bad thing, as it means that there are major differences in the pieces. The important part is the accuracy of the scores, which can be adjusted through our design. One thing that could further improve our design is extending the chord template to seventh chords and chords from the Armenian sharagran as discussed in chapter 2.1. A flaw in our current algorithm is that suspensions, passing tones, and the like are not accurately detected. Although these can easily be determined by examining a manuscript, it is not as easy to implement through computational design. Alternative approaches to our current methods can also be considered. For example, instead of HPS, we can instead attempt audio segmentation through non-negative matrix factorization, or NMF. This can be done by gradient descent, where we wish to minimize the Euclidean distance between a templates matrix, an activations matrix, and a spectrogram-like matrix, all which detail musical properties.

Our study could also be expanded to examine other musical aspects that aren't directly clear from our spectrogram/chromagram-based algorithms. Some expansions of our analysis include beat tracking to determine tempo and structure analysis to divide the music into sections. This can demonstrate non-harmonic differences by displaying differences in speed and form.

Although music can sound very related to one another, there are significant underlying differences that make them feel much different. The changes in harmony, timbre/instrumentation, and tempo all play roles in changing the context and interpretation of a melody. What we have found is that despite our ears picking up similarities, the truth is that the songs can be very much different. This can be proven pictorially, analytically, and quantitatively.

Pictorially, we saw how spectrograms and chromagrams detailed the frequency content of a piece of music. Comparing these between songs, we can see how similar and different they are in terms of notes and their intensities.

Analytically, we first looked at dynamic time warping to detect which regions between two songs are most similar. We retrieved an optimal path between songs, which tells us what the best alignment of them are. Then, we divided audio into harmonic and percussive components, whose combination resulted in our original signal. Here is where we discovered how much percussive influence there was in the harmony. We then pictorially reviewed the separated spectrograms, showing us exactly how much substance there was in the harmonic and percussive components.

Quantitatively, a score formula was decided on and served as a measure for harmonic similarities. The more chords that were shared between two songs, the better they scored. Given that a similar melody was present in all of the songs, the score gave us a measure to determine which songs were harmonically similar and which were different.

Despite the similarities and differences, the spirit and impact of the melody remains. “Der Voghormia” continues to convey its powerful message throughout the many contexts given by its many renditions.

Appendix A: Reference Table

Name	Artist/Composer	Notes
Der_Voghormia_1 DV1 Appendix B	Gomidas Vartabed	Digitally generated audio Transcribed from physical manuscript Three tenors, one bass
Der_Voghormia_2 DV2 Reference [10]	System of a Down	Professional recording Discography name: “Arto” Duduk, Percussion, Vocals
Der_Voghormia_3 DV3 Appendix B	Catholicos Simeon	Digitally generated audio Transcribed from physical manuscript Voice + Piano
Der_Voghormia_4 DV4 Reference [5]	Eileen Khatchadourian	Live Performance Band: voice, drums, strings, electric guitar incorporates a funk groove
Der_Voghormia_5 DV5 Reference [9]	Armenia Sarkissian	Professional recording Medley arrangement of 3 different variations

Appendix B: Digital Manuscripts

Excerpt of the first verse from Gomidas Vartabed's version of "Der Voghormia"
(A. Alaverdyan, personal communication, March 18, 2022)

Digital manuscript of Simeon I's version of "Der Voghormia"
(G. Saraydarian, personal communication, April 10, 2022)

Appendix C: References

1. Capan, L. (2005, August 15). *Sacred songs of the Armenian Apostolic Church*. https://www.anestesianet.com/capan/
2. Driedger, J. (2016). *Processing music signals using audio decomposition techniques* [Doctoral dissertation]. University of Erlangen–Nuremberg.
3. Grosche, P. (2012). *Signal processing methods for beat tracking, music segmentation, and audio retrieval* [Doctoral dissertation]. Max Planck Institute for Informatics.
4. Kerovpyan, A. and Kerovpyan, V. [Armenian Institute]. (2020, June 24). *Music with Aram and Virginia Kerovpyan: Musical modes and the Armenian eight-mode system* [Video]. YouTube. https://www.youtube.com/watch?v=XEY-JVq3BBM
5. Khatchadourian, E. [Eileen Khatchadourian]. (2015, April 29). *Eileen Khatchadourian - Der Voghormia (live at the In Memoriam Concert)* [Video]. YouTube. https://www.youtube.com/watch?v=P8Xpn6Luc-U
6. Liu, X. (2019). *Hidden Markov models for music classification* [Honors thesis]. Wheaton College.
7. Müller, M. (2015). *Fundamentals of music processing*. Springer.
8. Rosenzweig, S. (2017). *Audio processing techniques for analyzing Georgian vocal music* [Master's thesis]. University of Erlangen–Nuremberg.
9. Sarkissian, A. [Armenia Sarkissian]. (2017, April 23). *Der Voghormia - Armenia* [Video]. YouTube. https://www.youtube.com/watch?v=BGkNS2H4kOA
10. System of a Down. [System Of A Down]. (2014, November 8). *Arto* [Video]. YouTube. https://www.youtube.com/watch?v=qNG7H0r2-SE
11. Vagramian-Nishanian, V. (1981). Gomidas Vartabed: Pioneer of Armenian folk music. *The Choral Journal 22*(3), 5-8.