Algorithmic Pitch Mapping and Chord Structure Analysis of Music

Evan Azevedo
Department of Physics, University of California, Santa Barbara, CA 93106

(Dated: April 13, 2017)

Musical structure, when not understood implicitly by the listener, is expressed in the languages of rhythm, pitches, and chords. Transcribing music in terms of pitches and chords can be done automatically by application of the Discrete Fourier transform and certain analytic methods. This data can then be collected and analyzed to reveal patterns within artists, identify genres, facilitate learning the music for instrumentalists, and more.

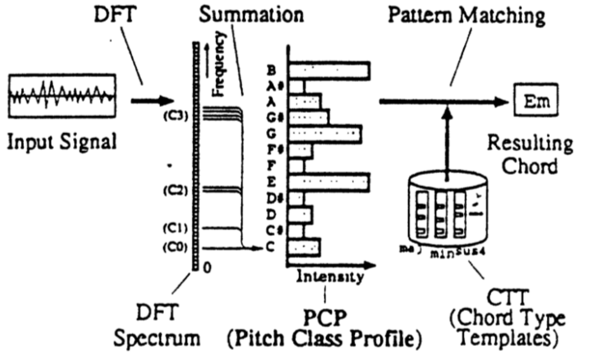


FIG. 1. A schematic of the Fujishima's pitch mapping and chord recognition algorithm. Using this method he was able to recognize chords to 94% accuracy in ideal conditions.

I. INTRODUCTION

The building blocks of Western music are 12 chromatic tonal steps, labeled A through G, with half intervals in between some, but not all, of the letters. Each of these tonal values has a traditional frequency associated with them, the most central of which being $A_4 = 440$ Hz. The approximate tonal structure of an audio file can be extracted from the Fourier transform of the signal. The Discrete Fourier transform is used to convert the spectrogram into a histogram of pitches over a small interval. The small intervals are then compiled and analyzed for their overall chord structure. The end result of this method is to represent the tonal and chordal structure of a song or segment of a song as a series of histograms of small time intervals, as seen in Fig. 3.

The application of this is to remove the need for human expertise in identification of notes and chord structure. Large databases can then be scraped for data, allowing for analysis and comparison of mood, style, rhythm, and general tonal development of musical pieces. More lucratively, this can also facilitate learning pieces without transcribed music. Fig. 1 shows a general schematic of the algorithm.

II. ANALYZING THE SIGNAL

The Fourier transform proves to be particularly useful in this method. It transforms the waveform function from the time domain $f(t)$ to the frequency domain $F(\omega_k)$[1]. The continuous Fourier transform of $f(t)$, $F(\omega_k)$, for $\omega_k = 2\pi k\nu$ with frequency ν,

$$F(\omega_k) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i k t}dt. \tag{1}$$

In our case, $f(t)$ is the input signal. Unfortunately, even on a low sampling rate this is very computationally intensive for an audio file. Even simple recordings are constantly changing in waveform and cannot be said to follow any pre-prescribed structure. Fortunately, the Fourier transform can be evaluated on discrete intervals. This is also within the interest of describing the song as a collection of musical phrases.

First, the audio signal is converted from a continuous stream of data into segments of periodic functions defined by a Discrete Fourier transform. This is done over a short time frame, $t \in [1 - N/2, N/2]$. The DFT (Discrete Fourier Transform), $F(\nu)$, from Lenssen's definition of the Fourier transform [1]

$$F(\nu) = \sum_{n=-N/2+1}^{N/2} f_n e^{-2\pi i k n/N}. \tag{2}$$

Note: we do not have to evaluate this sum across the whole range $n \in [0, N]$ because real signals are symmetric about DC voltage and the negative signal is redundant.

Computing time can further be reduced in this formula using the Cooley-Tukey algorithm, or the Fast Fourier transform. This cuts the computing time from n^2 to $nlog(n)$. To get an idea of how much faster this is, Lessen's paper uses the example of a standard 3 minute song, with a reasonable sampling frequency of 11025 Hz. That is an evaluation of about $2 * 10^6$ input points, and thus the FFT is $\sim 1,000\%$ faster [1]

$$\frac{n^2}{nlog_2 n} = \frac{2 * 10^6}{(2 * 10^6)log_2(2 * 10^6)} \approx 100,000. \tag{3}$$

The resulting waveform is then mapped onto the Pitch Class Profile domain, classifying the tone based on its maximal frequencies [2]. Each PCP is represented by a histogram of Pitch Classes ranked 0-11 for each pitch in Western music.

A. Mapping to the Pitch Class Profile

From this result, now the PCP can be identified. For some PCP, PCP(p) where each integer value of $p = 0, 1, ..., 11$ represents one of the defined pitches in Western music (A, A#, B, etc.), then $p(k)$ matches the spectrum bin to the PCP index. For example, if p=0 represents the pitch A and the frequency A=440 Hz, the p(k) will map $\|F(k)\|^2$ indicating the peak at 440 Hz to the pitch A. That is PCP is the sum of the transforms of the matched pitches over its interval.

Fujishima's method of classifying the PCP begins by mapping the k bins from the DFT to the PCP bins p, such that [2]

$$p(k) = round(12 \log_2 \frac{f_s/N}{f_{ref}}) \mod 12 \tag{4}$$

where the $12 \log_2$ product is rounded to the nearest integer value and f_s/N represents the frequency of the spectrum bin $F(k)$. For $N/2$ frequency points identified in an audio signal, then the width of frequency bins [3]

$$\Delta f = \frac{f_s}{N} = \frac{1}{N \Delta t} \tag{5}$$

where Δt is the time interval sampled. The Δf between pitches for western music increases as the pitch increases, resulting in larger spectrum bins for higher pitches. For example, Δf between D_1 and $D_1^{\#}$ for first octave of D is 2.18 Hz. Fujishima used the sampling frequency 8.0 kHz with a DFT length $N = 2048$, resulting in frequency bins of length ~ 4 Hz.

The PCP is then the magnitude of the signal in frequency bin k mapped to the pitch class p; [4]

$$PCP(p) = \sum_{k:p(k)=p} ||F(k)||^2. \tag{6}$$

This outputs a twelve dimensional vector $PCP(p) = (A, A^{\#}, B, ..., G, G^{\#})$ where each dimension is the magnitude of the corresponding pitch.

This method for identifying tones was pioneered in Fujishima's paper [2], and is the basis for pitch mapping for most modern methods. Now chords can be extrapolated in continuation with his method by comparing the values of the most prominent pitch classes over the interval to a bank of chord models.

Group	Chord Types
S1	$G_{single}, G_{dim}, G_{dim7}, G_{m7}^{-5}, G_{dimM7}, G_m, G_{m7}, G_{mM7}, G_7, G, G_{M7}, G_{aug}, G_{aug7}, G_{augM7}, G_{sus4}, G_{sus47}$
S2	$G_{m9}, G_7^{-9}, G_9, G_7^{+9}, F_m/G, F_{mM7}/G, F/G, F_{M7}/G, F_{dimM7}/G, F_{augM7}/G, G_{M9}$

TABLE I. An example of CTT for chords with a G root

B. Pattern Matching

Chords in Western music are defined in relation to the "root", or the basis of the chord, and the intervals of the other tones in the played at the same time from that root. In the chord matching method, data of the PCP over a small interval is now compared to the CTT (Chord Type Template) which gives the overall chord type of that interval. For example, an octave of the C-Major scale is (C, D, E, F, G, A, B, C) and a simple C-Major (C_M) chord has the $1^{st}, 3^{rd}$, and 5^{th} notes in that scale. Written in the format of data from the PCP, a C_M chord looks like Equation 7,

$$\begin{aligned} CTT(p) &= (A, A^{\#}, B, C, C^{\#}, D, D^{\#}, E, F, F^{\#}, G, G^{\#}) \\ &= (0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0). \end{aligned} \tag{7}$$

Table I shows an example of the possible CTT's for chords rooted at G, taken from Fujishima's paper. Group $S1$ are the chord types that are defined within one octave, and group $S2$ are chord types that span over an octave. The amount of chords in these groups is a higher level of precision than this method can reasonably match. When applied to a real world case, this method is better suited for a bank of 3-7 chords [4]. This can be done by removing chords that are very unlikely to be seen contextually. Sheh and Ellis' response to this was raising the resolution of the PCP to be a 24 dimensional vector, rather than 12, allowing for more precise chord modeling [4].

One problem others have encountered with chord mapping is unfortunately prevalent in nearly all forms of music: percussion. Percussive instruments have many overtones and harmonics, but they are also tuned and produce leading tones that, due to their volume relative to other instruments, can overpower chord progressions and indicate roots of chords that should not be there. Fujishima's method includes ignoring the base tone PCP(0) as an analytic solution to this problem. However, statistical modeling can effectively learn conditions for the context of chords and improve accuracy in chord mapping.

In Christopher Raphaels paper, titled *Automatic Transcription of Piano Music*, he outlines his use of Markov Modeling [5], a principle of machine learning, to quickly identify and account for variables such as percussion, room acoustics, signal noise, and other features of musical recording that could distort transcription of the signal [6]. Markov Modeling, while not necessarily related to the Fourier transform or complex analysis, is a powerful tool in vocal recognition and has been shown to have promising applications in this field.

C. Implementation of HMM

Building upon Fujishima's method of pitch mapping and chord templates, collected data can be used to train a Hidden Markov Model, or HMM. The HMM is "a stochastic finite automation in which each state generates an observation" [4]. In other words, the messy, inconsistent data from the PCP(p) values can be modeled and refined by a statistical process which recognizes patterns in the chord data, and tests predictions to sharpen its results.

We define X as the observed feature vectors (in our case the PCP), Q as the unknown chord labels assigned to the PCP, and Θ as the current model parameters. Then the Expectation Maximization (EM) algorithm, of the form $E[P(X)]$, is:

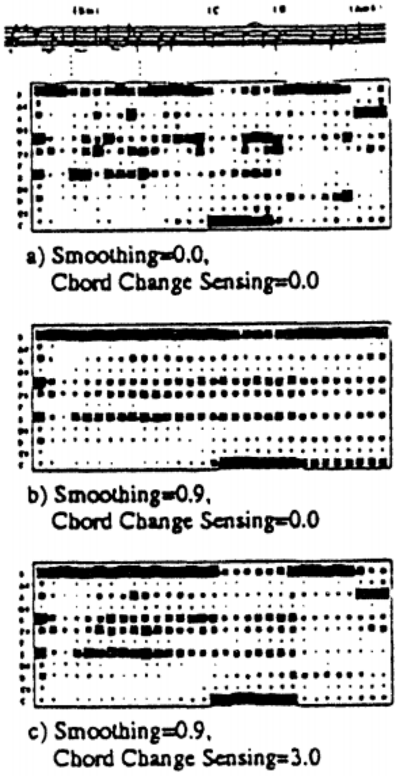
$$E[logP(X,Q|\Theta)] = \sum_{Q} P(Q|x,\Theta_{old}) log(P(X|Q,\Theta))P(Q|\Theta). \tag{8}$$

This algorithm expresses the data our total song segment with relation to two parameters: our old model Θ_{old}, and new Θ. At each step Q, Θ is evaluated and replaces Θ_{old} in the next step until the difference between the two is some small ϵ [4].

More specifically, EM training gives a set of model parameters for each defined chord states. It will tell us information about how the A_{M7} chord, for example, is used throughout the data set. Sheh and Ellis built upon the use of this algorithm by calculating the weighted average of the models for each chord family across all of the roots [4]. This technique effectively normalizes each chord training set in the model and showed a significant improvement in their results.

III. RESULTS

Fujishima's method, while sturdy, has the potential to collect some pretty messy data, as seen in (Fig. 2). His method for clearing up the data is a little simpler than either Rabiner's or Sheh and Ellis' HMM methods by comparison. He hand tuned his data by two factors, "Smoothing" and "Chord Change Sensing", and adjusted their values to yield the clearest data. The Smoothing parameter, for example, reduces noise, but it also "oversmooths, blurring chord change points" [2]. His paper includes the chord reading of this same piece and by using his heuristic parameters and a chord table chosen for recorded music (rather than keyboard sounds) he was able to reproduce chord fitting with an accuracy of 94% [2].



a) Smoothing=0.0,
Chord Change Sensing=0.0

b) Smoothing=0.9,
Chord Change Sensing=0.0

c) Smoothing=0.9,
Chord Change Sensing=3.0

FIG. 2. Fujishima's method involved tuning factors by hand to make the transcription more precise. The heuristics show an increase in consistency from a) to b) and accentuation of structure in c)

Nathan Lenssens paper is more accessible for the average reader and is a more holistic description of musical analysis and its methods. He presented a pitch mapping of a 12 second clip of chromatic tones played on a piano [1]. The data had a clear pattern of periodically raising pitch and evidently represented the sounds he described. Considering that his paper was written in application for a scholarship, rather than research purposes, his results are a

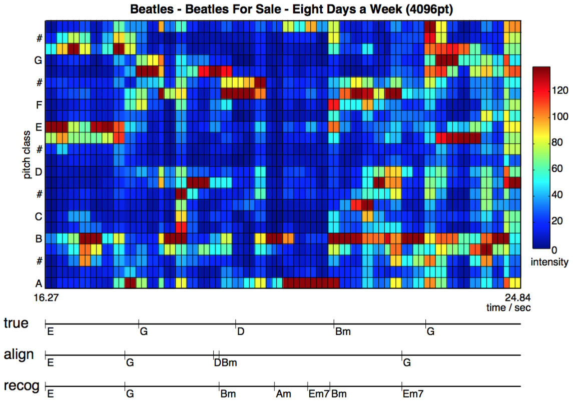


FIG. 3. HMM chord modeling of an 8 second segment of Eight Days a Week, showing the recognized and forced alignment results versus true chord structure

formidable proof of concept for pitch mapping and could be built upon to include chord structure.

Raphael's paper focuses on the transcription of solo piano music. By limiting his scope to a single instrument he was able to make steps towards sheet-music level of transcription, distinguishing between flux and steady state behavior, or "attack" and "sustain". He measured his success by the "note error rate", calculated by the minimum number of insertions, deletions, and substitutions needed to yield the true note sequence from the recognized one. His method found a note error rate of 39% on transcription of a Mozart piano sonata [6]. While this may seem trivial compared to Fujishima's highest success rate of 94%, Raphael's result is important because he calculated his error on a note by note basis and therefore was able to recognize music successfully at a much higher resolution than Fujishima.

Sheh and Ellis [4] were able to successfully identify chords and tones in several popular Beatles songs using the expectation maximization algorithm (Fig. 3). They trained the HMM in two steps: forced alignment, and recognition. By forced alignment the chord sequence is given and the model gives predicts where the chord changes happen. In recognition the HMM is not constrained to any progression or special set of chords and will perform quite poorly in accuracy without more information. The improvement from recognition to forced alignment as seen in Fig. 3 indicates that the model has improved accuracy by applying the chord characteristics. At the bottom of Fig. 3 are the true chord progression, the forced alignment results, and the recognition output.

IV. DISCUSSION

The property of automatic transcription to be quicker and scalable to larger data sets than possible with manual transcription shows potential given the demand for transcribed music. The methods described in this paper all accomplish this task in different ways and with different degrees of success. Fujishima's introduction of the PCP and CTT is the basis of the other methods described in this paper. He found that he could correctly identify chords, even in complex orchestral pieces with relative accuracy [2]. His foundation for data collection is augmented by Sheh and Ellis by using HMM and statistical methods to improve chord recognition. Raphael's work focuses on solo piano pieces and tailoring the process of HMM for higher single note recognition.

When written music for a piece is not found online, chord recognition websites will often be of the first search results. For example, the website *http://www.riffstation.com* shows how the chord progression of a song on Youtube changes while the song plays. The website's chord recognition algorithm detects major, minor and 7th triads - notably more restricted than Fujishima's or Sheh and Ellis' methods. A test with the song *You Know What To Do* by the Shivas shows the tenancy of percussion instruments to have a hide chord changes and overpower the rhythm guitar.

Despite results neither as precise nor as accurate as shown possible by Raphael or Fujishima, the website has messages indicating that it is unable to process the amount of queries it receives. Music recognition is simply not as developed as voice recognition, despite having similar problem sets. With the right implementation, automatic chord transcription has potential to be a useful tool to music learners, studiers, as well as valuable to data science in general.

V. ACKNOWLEDGEMENTS

I would like to thank Henry Taylor and Sachin Guliani for their constructive criticism and helpful insight on this paper. Also thank you to Dr. Jean Carlson for a challenging and constructive course.

[1] Nathan Lenssen, "Applications of Fourier Analysis to Audio Signal Processing: An Investigation of Chord Detection Algorithms," CMC Senior Thesis , 253–291 (2013).

[2] Takuya Fujishima, "Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music," CCRMA , 464–467 (1999), arXiv:1503.06237 [hep-th].

[3] "The Fundamentals of FFT-Based Signal Analysis and Measurement in LabVIEW and LabWindows/CVI," National Instruments (2009).

[4] Alexander Sheh and Daniel P.W. Ellis, "Chord Segmentation and Recognition using EM-Trained Hidden Markov Models," Proc. Int. Conf. Music Information Retrieval , 1113–1133 (2003).

[5] Rabiner R. Lawrence, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," **77**, 257–286 (1989).

[6] Christopher Raphael, "Automatic Transcription of Piano Music," Univ. of Mass., Amherst , 110501 (2002).