# Automated Music Transcription

Tejas Sodhi
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai,India

Sayed Abdur Rehman
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai,India

Yash Rajput
Department of Computer Engineering
VidyalankarInstituteofTechnology
Mumbai,India

***Abstract :*** In this paper, we are proposing the idea of making an automated software that will transcribe each note while the musician plays the instrument. The software will take the sound of the instrument as an input and will process the frequency of each note and the way it is played and transcribe the note visually. In this project, we will consider recording consisting of only monophonic notes i.e. only a single note will be played at a time.This paper focuses on extracting audio, detecting pitch and displaying symbols. The project makes extent use of audio signal processing and python libraries (pyAudio)..

***IndexTerms - audio signal processing, pyAudio.***

## I. INTRODUCTION

Composing music is form of expressing an artistic mind. The composed musical pieces are transcripted using the musical language. Noting down these transcript can be a very difficult task as a piece/song may have lots of notes and each note must be written carefully with all the articulation and timing properly represented. Also it may take up a lot of time writing down each note on a sheet manually. This might hinder the creative aspect of writing/composing music.

This paper concerns automating the music transcription process which converts music recording into symbolic representation. In order for a computer to be able to do such a task, it must take a musical input from a sound file, usually of the .wav variety (from MIDI input) and perform an analysis of frequency and duration. In this paper, we propose to generate transcriptions through 3 step process.

The system records either a live audio or a prerecorded one and stores in a file. This file is then passed to the pitch detector code where it is analyzed. We used FFT to detect the peak frequency of an audible audio. FFT is apt for this project for its speed enhancements and is capable to take a waveform in time domain and convert it into frequency domain.

FFT acts like a tuner on an individual time sample; pitch is returned at that specific time input. The resultant frequency values are then noted down and compared with the frequency values of musical notes. Each frequency corresponds to a note and as per the input audio signal, corresponding symbol is generated. The resultant sheet is displayed in Lilypond with the help of Music21

## II. LITERATURE SURVEY

[1] TECHNIQUES FOR AUTOMATIC MUSIC TRANSCRIPTION by Juan Pablo Bello, Giuliano Monti and Mark Sandler

This paper introduces to parameters needed to be analyzed in music transcription.It also stated the methodology of monophonic transcription using autocorrelation pitch tracking and implementation of segmentation as well as neural networks.

[2] Automatic Music Transcription as We Know it Today by Anssi P. Klapuri Tampere University of Technology, Tampere, Finland

This paper cites various problems associated with music transcription and proposes solution to it by using various audio spectrum analysis. It also describes properties of musical sounds and the basic principles of F0 estimation, different approaches to multiple-F0 estimation will be introduced. Work on rhythmic parsing and future prospects of music transcription.

[3] Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence by Arnaud Dessein, Arshia Cont, Guillaume Lemaitre

This paper proposes the solution for polyphonic music transcription using a modified non-negative matrix factorization scheme, called non-negative decomposition, where the incoming signal is projected onto a fixed basis of templates learned off-line prior to the decomposition.It also discuss the use of non-negative matrix factorization with the β-divergence to achieve the real-time decomposition.

[4]A Generative Model for Music Transcription by Ali Taylan Cemgil Bert Kappen David Barber

In this paper we are presented with a graphical model for polyphonic music transcription. The model, formulated as a Dynamical Bayesian Network, embodies a transparent and computationally tractable approach to this acoustic analysis problem.

## III. PROPOSED METHODOLOGY

- *AUDIO RECORDING*

For performing music transcription, it is necessary to have audio file to transcribe. The audio files used in this project will be in .wav file format. .wav files contain the necesssary waveform data to check for pitch and duration. After choosing the .wav audio file to transcribe, it is necessary to make sure it is free of metadata.

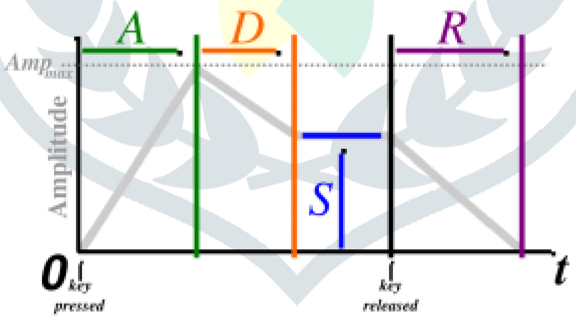
Metadata is basically data that stores information about another set of data. This needs to be removed. To clear the metadata, we can make use of a software called Audacity. Now that the metadata has been cleared away, data can be extracted. The tool used to do this is Scipy's own waveform data extractor; **scipy.io.wavfile.read().**

The result is an array containing the sampling rate and waveform data. Normalisation may be performed to remove any anamolies that may affect an effictive mapping of data. The waveform needs to be continuous one else an anomaly called spectral leakage will occur. Spectral leakage will cause a spill of audio data into other bins, thus creating an excess amount of noise; the FFT will be flooded by this noise and it would hinder the working. We have performed normalization by utilizing Scipy's Hann function, **scipy.signal.hann()**. The data set will become windowed and appear continuous to the FFT, making for better accuracy in pitch estimation.

- *PITCH DETECTION*

After normalizing the waveform audio data, it is possible to extract pitch. The simplest way to determine the frequency of a note would be to compute the spectrum of sound signal by taking the Fast Fourier Transform (FFT) and finding the frequency at which we get the maximum amplitude. This function takes input in the time domain and will output coefficients in the frequency domain. The input data from the .wav file is in the time domain.

Essentially, this maps waveform audio data to a discernible frequency, otherwise known as pitch. Due to issues with noise, only the peak value of each index or the audible data will yield the appropriate pitch. Additionally, to ensure greater accuracy, the FFT will be performed on smaller subdivisions of the sound file. These subdivisions are called as chunks. These chunks allow the algorithm to even consider minute pitches to determine values that otherwise might miss out. Chunks are necessary for this mapping as otherwise, the FFT would only return one pitch value; the more chunks there are, the more data will be mapped to pitch. This is the basic of sampling .Every element returned from the FFT is then linked directly to a known pitch value, in the form of the previously mentioned frequency file.




This file is .txt document that stores all the known frequency values. Using python, this frequency file is read into a dictionary. Then, an array of equal size to the FFT result is created whose elements are equated to the closest absolute pitch from frequency to the corresponding coefficient in the FFT output.

Each index represents a point in the audio file and contains the corresponding pitch, as given by the FFT. To appropriately place our music sheet on staff we need to consider facets like tempo, note length and rests. To provide this, we make use of Bayesian networks. It recursively runs the mapped pitch data and compares the result .

- *TRANSCRIPTION*

The final task is to transcribe the audio file to digital sheet music. Here we make use of Music21.It is a free online program that will allow for this transformation. All that must be done is open a stream object from Music21 in python and then append all our notes; each index in the pitch and rhythm arrays correspond to a note.

To append, the array will be looped over at each index, creating a note whose pitch and duration map the values of the corresponding rhythm and pitch indexes. Upon completion of the loop, a virtual staff will exist containing notes derived from the initial waveform audio file.

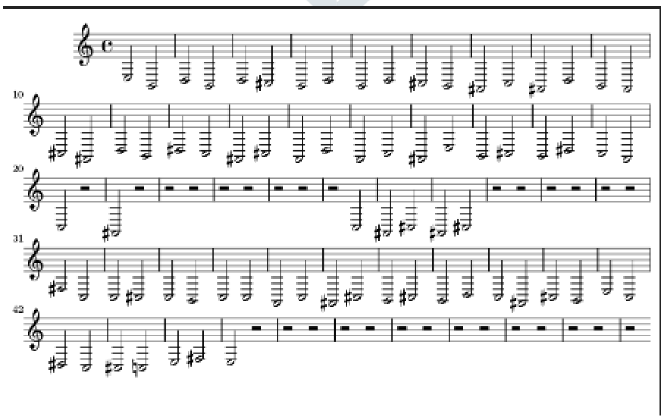
Music21 is unable to display sheet music on its own, so the program will enlist the help of Lilypond to engrave the stream as a digital image. By calling a show("lily") method, Music21 is capable of completing the transcription process; data gathered by manipulating audio is finally put done in a standard notation that a musician can read.

## IV. RESULTS AND DISCUSSION

1. Twinkle Twinkle Little Star


2. ode to joy


3. 7Nation army



## V.CONCLUSION

This paper proposes to automate something that is currently being done manually. It attempts to sample out chunks of pitch samples, compare with the actual referenced pitch values and display the transcribed symbols on an interface platform. The above results proves the accuracy to be over 80 percent. The project includes various technologies to complete the given task. It heavily relies on FFT and python for its successful completion. Use of music21 along with lilypond is used to provide a smooth music sheet over a staff.

## VI.FUTURE SCOPE

In future we can try to improve the efficiency of the transcriptions. We will also try to work on chords and multiple notes

## VII.ACKNOWLEDGMENT

For publishing this paper, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. We would like to show our gratitude to Dr. Arun Chavan, Head of Computer Department, V.I.T. for his useful insights. We would also like to thank Prof. Devendra Pandit for giving us a good guideline throughout numerous consultations.

## REFERENCES

[1] Bacon, R. A., Carter, N. P., and Messenger, T. “The Acquisition, Representation and Reconstruction of Printed Music by Computer: A Review.” Computers and the Humanities. Vol.22, No.2 (1988): 117 – 136. JSTOR. Web. 9 March 2012.

[2] Bello, Juan Pablo, Monti, Guliano and Sandler, Mark. “Techniques for Autmoatic Music Transcription.” King's College London. Web. 9 March 2012.

[3 ] Cheng, Xiaowen, Hart, Jarod V., and Walker, James S. “Time-frequency Analysis of Musical Rhythm.” PDF File. Web. 2 Jan 2013.

[4] TECHNIQUES FOR AUTOMATIC MUSIC TRANSCRIPTION by Juan Pablo Bello, Giuliano Monti and Mark Sandler.

[5 ] Automatic Music Transcription as We Know it Today by Anssi P. Klapuri Tampere University of Technology, Tampere, Finland

[6] A Generative Model for Music Transcription by Ali Taylan Cemgil Bert Kappen David Barber

[7] ] Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence by Arnaud Dessein, Arshia Cont, Guillaume Lemaitre