Automatic Transcription of Piano Music

Shruti Rachh
Indiana University, Bloomington
srachh@iu.edu

Abstract

The goal of this project is to automate the music transcription process from a musical recording. In this project, we will consider piano recordings consisting of only monophonic notes i.e. only a single note will be played at a time. There are many algorithms to generate musical sheets from an audio such as autocorrelation, CQT transform and cepstral based. This paper will focus on YIN which is a modified autocorrelation algorithm and cepstral algorithm and will compare the performances of both the algorithms.

Keywords: YIN, autocorrelation, cepstrum

INTRODUCTION

This paper concerns automating the music transcription process which converts music recording into symbolic representation. Musicians usually manually write down the notations on a piece of paper while composing music. So automating this task will help save a lot of time. In this paper, I propose to identify the pitches in the recording using two different methods and will then provide a comparison of the two. The first method is based on YIN algorithm which is a modified version of autocorrelation, but with greater accuracy. The second method uses cepstral based pitch detection. I will be using the recordings played on a piano and it is assumed that the audio contains only monophonic pitches, that is, only one note is being played at a given time.

The process of transcribing music consists mainly of two steps:

1. Onset detection
2. Pitch detection

Each of these steps are discussed further.

APPROACH

1. Onset detection

Onset detection is used to determine the locations where each note begins. This can be done as follows:

a. Divide the sound into frames of fixed window size (W) such that either one note or noise can be detected within each frame.

b. For each frame, compute the spectral energy of the signal given by:

$$intensity_t = \sqrt{\sum_{j=t}^{t+W} x_t(j)^2} \text{ -- (1)}$$

Where $intensity_t$ is the energy of the t th frame

c. Compute the ratios of intensity of each frame and its previous frame. Greater the value of this ratio indicates the beginning of the new note. Compare these ratios with a threshold, if greater than that, then save that frame number and send it to pitch detection to determine the fundamental frequency. Determining the threshold that works for all the recordings is a challenging task.

2. Pitch detection

Pitch detection can be done using several methods:

FFT

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. But the problem with this method is that it is not necessary that we would always get the fundamental frequency, we may get the harmonic of the fundamental frequency instead.

Cepstrum

Computing the cepstrum would give frequencies f, f/2, f/3 and so on. The problem with FFT is not present when computing cepstrum. So, we can successfully find the fundamental frequency corresponding to the maximum amplitude.

Product of FFT and Cepstrum

The product of FFT and cepstrum will give only the fundamental frequency since FFT gives harmonic frequencies f, 2f .. and cepstrum gives frequencies f, f/2.. . This method will also work well, in fact, better than the cepstrum method, but, it is difficult to compute. The difficulty comes with the indices at which we get frequencies 2f, 3f in FFT may differ from the indices where we get frequencies f/2, f/3 so taking the product directly without considering this fact will give incorrect results.

YIN algorithm using autocorrelation

Autocorrelation is the method used to compare the similarity of the signal with its delayed version. This method also correctly computes the fundamental frequency but it is slower than the cepstrum method.

The focus of this paper will be on pitch detection using cepstrum and YIN algorithm and are discussed further.

YIN algorithm

The YIN algorithm is based on the autocorrelation of the sound signal but with some modifications which helps to prevent errors that are found with autocorrelation.

The algorithm is as follows:

a. Autocorrelation computation

The autocorrelation of signal is defined as follows:

$$x_t(\tau) = \sum_{j=t+1}^{j=t+W-\tau} x_j x_{j+\tau} \quad \text{-- (2)}$$

Where $x_t(\tau)$ gives the autocorrelation of the t th window frame of sound, W is window size and τ is the lag or delay between the signals. We need to determine the lag for which the autocorrelation $x_t(\tau)$ is maximum. The fundamental frequency can be computed as:

$$f = sr/\tau \quad \text{-- (3)}$$

Where sr is the sampling rate of the audio. But, this method will not always give the correct fundamental frequency, so instead of taking the product of signal with its delayed version, we can take the squared difference between the two and find the τ_{min} which indicates that the two signals are most similar.

b. Difference function

$$d_t(\tau) = \sum_{j=t+1}^{j=t+W-\tau} (x_j - x_{j+\tau})^2 \quad \text{-- (4)}$$

Where $d_t(\tau)$ gives the squared difference autocorrelation of the signal.

c. Normalize

Before determining the τ_{min} using equation (4), we first normalize $d_t(\tau)$ as follows:

$$d_t'(\tau) = 1 \text{ when } \tau = 0 \quad \text{--(5)}$$
$$= \frac{d_t(\tau)}{1/\tau(\sum_{j=1}^{\tau} d_t(j))} \quad \text{otherwise}$$

We now find the lag for which the above equation is minimum (τ_{min}) and then

compute the fundamental frequency using equation (3).

This algorithm takes a long time to run because for determining each pitch we need to loop through the window size of the frame and compute the difference function for each time lag τ within a certain range. The time complexity can be improved by computing autocorrelation using FFT method.

Cepstral algorithm

Cepstrum of a signal is computed as:

$$c(\tau) = FFT^{-1} \log(\text{FFT}(x)) \quad \text{-- (6)}$$

Where x_t represents the t th frame signal. The pitch can be determined by taking the peak of the resultant signal in cepstral domain within a certain range. The cepstrum is given in terms of quefrency (τ) which is the pitch lag. The frequency can then be determined using equation (2).

This algorithm works well and takes much less time as compared to YIN algorithm.

IMPLEMENTATION

Both the algorithms were tested for two music recordings "DG" containing only 2 notes D and G and "Mary had a little lamb" and the results are compared below:

1. DG

The actual notes played in recording are: D G

Onset Detection: The below graph shows the ratios of intensities vs frames of song for the recording "DG". When the ratios are compared with the threshold we get the location (frame number) corresponding to start of each note.

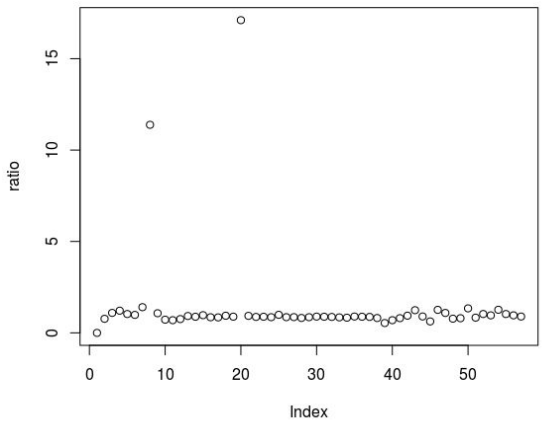

Figure 1: Onset detection (ratio of intensities) for "DG" recording

As seen, it correctly detects the onset of both the notes having ratios approximately 12 and 17 respectively.

YIN algorithm:

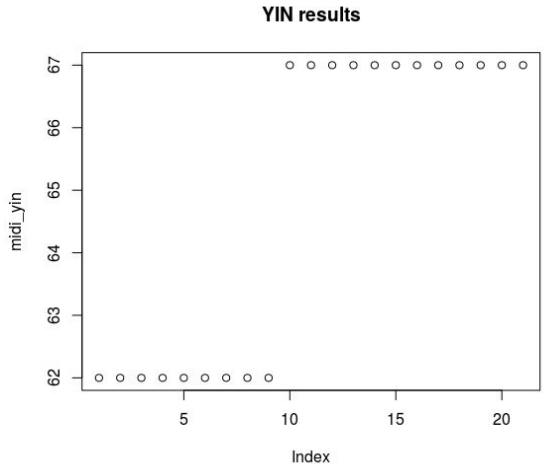

Figure 2: Pitch detection using YIN algorithm for DG recording

The above figure shows the pitches detected in recording DG using YIN algorithm. It shows the MIDI value of the notes in each frame of the recording. MIDI value of 62

corresponds to D note and 67 corresponds to G note which is correct.

Cepstral algorithm

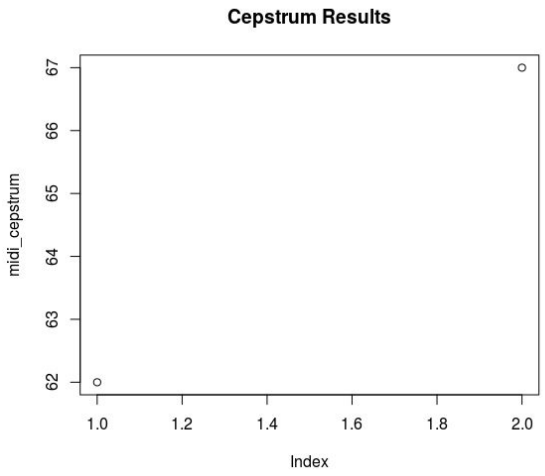

Figure 3: Pitch detection using cepstral algorithm for DG recording

The above figure shows the pitches detected in recording DG using cepstral algorithm. This algorithm also correctly detects D and G notes. MIDI value of 62 corresponds to D note and 67 corresponds to G note.

2. Mary had a little lamb

The actual notes played in recording are:
E D C D E E E D D D E E E

Onset Detection: The graph in Figure (1) shows the onsets of different notes for the recording "Mary had a little lamb".

YIN algorithm:

The figure (2) shows the pitches detected in recording DG using YIN algorithm. The top portion of the graph shows correctly detected notes, but there are some noisy notes that are getting detected at the bottom.

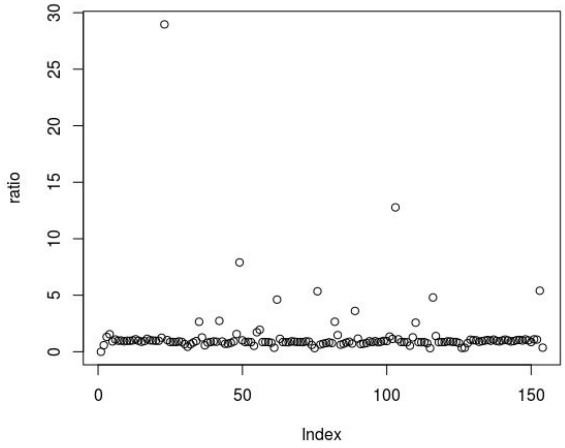

Figure 1: Onset detection (ratio of intensities) for "Mary had a little lamb" recording

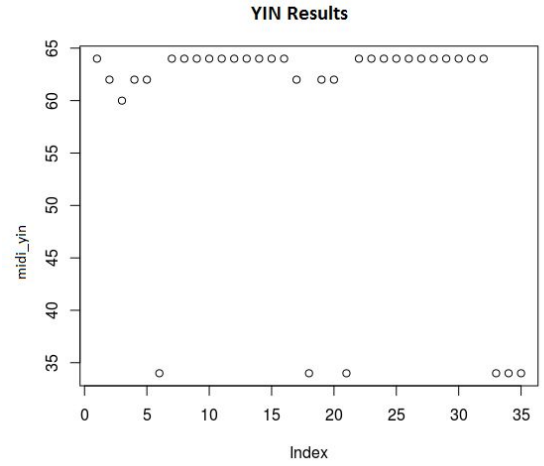

Figure 2: Pitch detection using YIN algorithm for "Mary had a little lamb" recording

Cepstral algorithm

The above figure shows the pitches detected in recording Mary had a little lamb using cepstral algorithm. As seen, only one noisy note is detected in the end at the bottom. So, the accuracy is much better than YIN algorithm and it is also faster to compute.

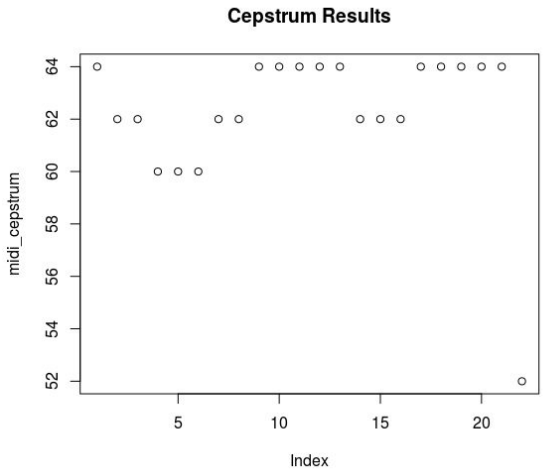

Figure 3: Pitch detection using cepstral algorithm for "Mary had a little lamb" recording

RESULTS

Song	Accuracy of YIN	Accuracy of Cepstrum
DG	100%	100%
Mary had a little lamb	82.85%	96%

FUTURE WORK

We can apply beat detection algorithms to detect the lengths of each note and the tempo of the song. With the knowledge of tempo, we can distinguish between when same note is played multiple times consecutively and when it is played only once which is not possible to determine right now. Once monophonic sound transcription is completed, it can be extended to detect polyphonic sound i. e. multiple notes being played at the same time.

CONCLUSION

Both YIN and cepstral algorithms were implemented for pitch detection in piano recording consisting of only monophonic notes. It was observed that cepstral algorithm was comparatively faster and accurate than YIN algorithm. Both of these algorithms will not be able to distinguish when a note is being played once, twice or multiple times consecutively.

REFERENCES

[1] Alain de Cheveigne, Hideki Kawahara: "YIN, a fundamental frequency estimator for speech and music", Acoustical Society of America, Vol. 111, No.4, 2002

[2] Kobayashi, H. and T. Shimamura (Sept. 1995). A weighted autocorrelation method for pitch extraction of noisy speech. In Proceedings of the Acoustical Society of Japan, pp. 343–344. Urawa, Japan: Saitama University

[3] A. M. Noll, "Cepstrum Pitch Determination", Journal of the Acoustical Society of America, Vol. 41, No. 2, pp. 293-309, 1967

[4] L. Rabiner ; M. Cheng ; A. Rosenberg ; C. McGonegal, "A comparative performance study of several pitch detection algorithms", IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume: 24, Issue: 5, Oct 1976

[5] Noll, M. (1969). Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In Proceedings of the Symposium on Computer Processing Communications, pp. 779–797. Polytechnic Institute of Brooklyn.