Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 57 (2015) 876 – 884

Procedia
Computer Science

3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)

Musical Notes Identification using Digital Signal Processing

Jay K. Patel[a], E.S.Gopi[a]

[a]National Institute of Technology, Trichy-620015, India.

Abstract

Songs play a vital role in our day to day life. A song contains basically two things, vocal and background music. Where the characteristics of the voice depend on the singer and in case of background music, it involves mixture of different musical instruments like piano, guitar, drum, etc. To extract the characteristic of a song becomes more important for various objectives like learning, teaching, composing. This project takes song as an input, extracts the features and detects and identifies the notes, each with a duration. First the song is recorded and digital signal processing algorithms used to identify the characteristics. The experiment is done with the several piano songs where the notes are already known, and identified notes are compared with original notes until the detection rate goes higher. And then the experiment is done with piano songs with unknown notes with the proposed algorithm.

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015
(ICRTC-2015)

Keywords: Time-Frequency Analysis; Musical Note; Sampling Frequency; Recording; Extraction;

1. Introduction

The ability to derive the relevant musical information from a live or recorded performance is relatively easy for a trained listener, but highly non-trivial for a learner and computer. For a number of practical applications, it would be desirable to obtain this information in a quick, error-free, automated fashion. This thesis discusses the design of a software system that accepts as input a digitized waveform representing an acoustical music signal, and that attempts to derive the notes from the signal so that a musical score could be produced. This signal processing algorithm involved include event detection, or precisely where within the signal the various notes actually begin and end, and pitch extraction , or the identification of the pitches being played in each interval. The event detection is carried out using the time domain analysis of the signal, where the problem arise with different speed. The pitch detection is (nothing but frequency identification) is more complicated because of a situation we call harmonic ambiguity; this occurs when one pitch whose fundamental

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)
doi:10.1016/j.procs.2015.07.499

frequency is an integer multiple of another pitch. The problem is solved by the careful signal processing in both the time domain signals and frequency domain signals.

The main objective of this project is to create an aid tool for learning for Musicians, Producers, Composers, DJs, Remixer, Teachers and Music Students. This project can be treated as a box, where you give any song as input and get the features of the song out. The aim of this project is to propose methods to analyze and describe a signal, from where the musical parameters can be easily and objectively obtained, in a sensible manner. A common limitation find in the musical literature is that the way in which such parameters are obtained is intuitively satisfactory but, to our view, not very sound from a signal processing perspective.

2. Literature Survey

2.1 Sound

A sound can be characterized by the following three quantities: (i) Pitch.(ii) Quality.(iii) Loudness.

Pitch is the frequency of a sound as perceived by human ear. A high frequency gives rise to a high pitch note and a low frequency produces a low pitch note. A pure tone is the sound of only one frequency, such as that given by a tuning fork or electronic signal generator. The fundamental note has the greatest amplitude and is heard predominantly because it has a larger intensity. The other frequencies such as 2fo, 3fo, 4fo,... are called overtones or harmonics and they determine the quality of the sound. Loudness is a physiological sensation. It depends mainly on sound pressure but also on the spectrum of the harmonics and the physical duration.

2.2 Musical Notes

Human can hear signal frequency ranging from 20-20 kHz. From this wide range some part is associated with piano. Different pianos are having different ranges. Each tone of piano is having one particular fundamental frequency and represented by a note like C, D, ...etc. as shown in fig 1 .The later C is 12 half steps away the previous one and having double the fundamental frequency. Hence this portion (from one C immediate next C) is called one octave. Different octaves are differentiated by C_1,C_2, etc.

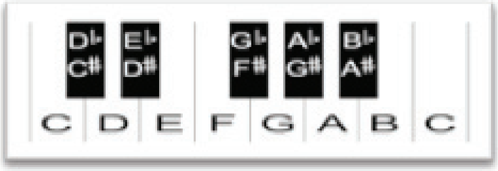

Fig 1. An octave of a piano

2.2.1 Equation For the Frequency Table

The basic formula for the frequency of the notes of the equal tempered scale is given by $f_n = f_0 * (a)^n$ where f_0= the frequency of one fixed note which must be defined. A common choice is setting the A above middle C(A_4) at f_0=440 Hz, n = number of half steps away from the fixed note you are, f_n= the frequency of the notes n half steps away, $a = (2)^{1/12}$

2.2.2 Frequencies for Equal Tempered Scale at A_4 = 440 Hz

Table 1. Notation to Frequency Mapping [Middle C is C_4]

n	Note	Fundamental Frequency(Hz)
-4	F_3	174.61
-3	$F_3^\#$	185
-2	G_3	196
-1	$G_3^\#$	207.65
0	A_4	220
1	$A_4^\#$	233.08
2	B_4	246.94
3	C_4	261.63
4	$C_4^\#$	277.18
5	D_4	293.66
6	$D_4^\#$	311.13
7	E_4	329.63

2.3 Digital Signal Processing for music

2.3.1 Sampling

When a sound wave is created by your voice (or a musical instrument), it's an analog wave of changing air pressure. However, in order for a computer to store a sound wave, it needs to record discrete values at discrete time intervals. The process of recording discrete time values is called sampling, and the process of recording discrete pressures is called quantizing. Recording studios use a standard sampling frequency of 48 kHz, while CDs use the rate of 44.1 kHz. Signals should be sampled at twice the highest frequency present in the signal. Humans can hear frequencies from approximately 20-20,000 Hz, which explains why common sampling frequencies are in the 40 kHz range.

2.3.2 Frequency and Fourier Transforms

A Fourier transform provides the means to break up a complicated signal, like a musical tone, into its constituent sinusoids. This method involves many integrals and a continuous signal. We want to perform a Fourier transform on a sampled (rather than continuous) signal, so we have to use the Discrete Fourier Transform instead.

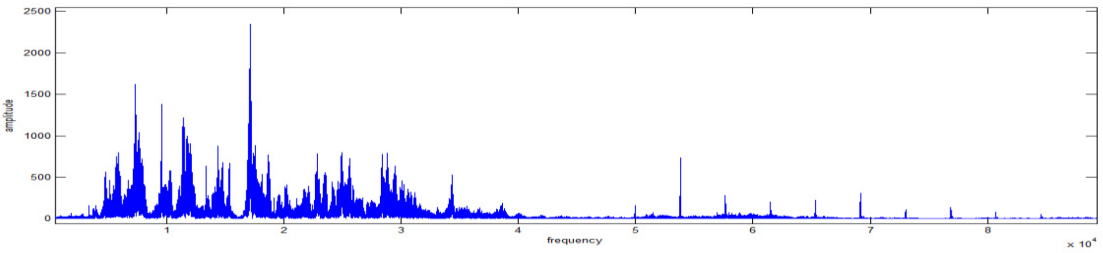

Fig 2. FFT of a Musical Signal

From the fig 2 we can see that the index corresponding to the maximum amplitude represents the most significant frequency component, that can be found using the formula

$$f = \frac{i}{T} * fs \tag{1}$$

Where i = index at which maximum amplitude exists, T = Total samples in the fft at a time.

2.3.3 Padding with Zeroes

Padding with zeroes, though a standard practice and useful in many applications, does not appear to significantly improve our data in this application. Though we have twice the frequency resolution, it's not yielding any better data. Quick tests padding with many more zeroes (1 part sample, 9 parts zeroes) show that though the peaks get rounder due to better frequency resolution, the discrepancy between the highest point on the original sample and the highest point on the padded sample is 1 Hz at most, which means that it is probably not worth the effort, even at low frequencies.

3. Methodology

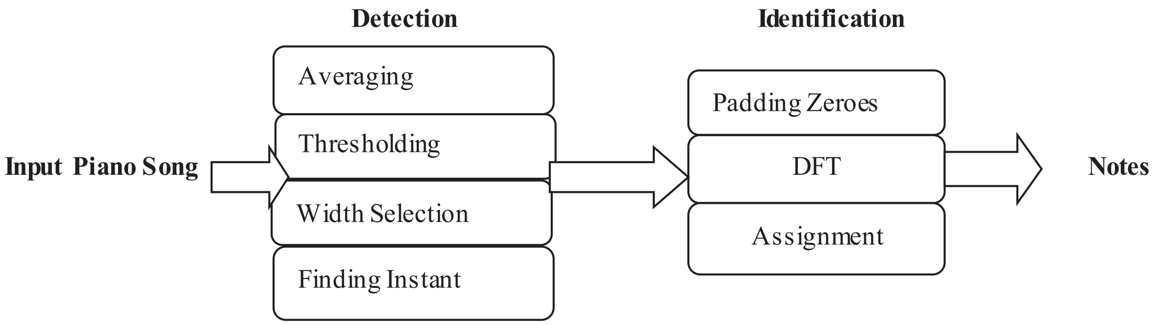

Fig 3. Flow-Chart

The experiments are done on different songs, some downloaded and some recorded from the virtual piano. Here each note follows a kind of similar pattern as shown in the fig 3.

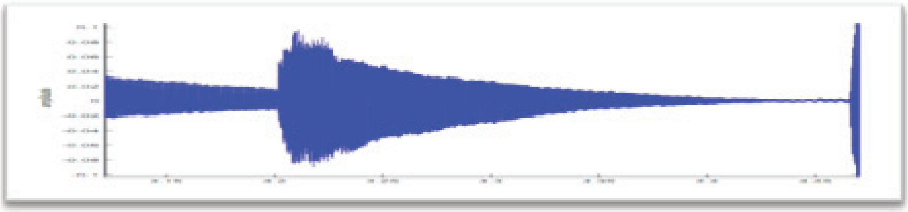

Fig 4. Piano note time domain analysis

The moment we press one note to the immediate other note, the amplitude is initially high enough and decreases with time. If we are able to detect the duration of each note from the time domain characteristics, we can detect and identify the frequency.

3.1 Detection

3.1.1 Averaging

As there is large number of sample for a song and many fluctuations are also present, first step is averaging, where for every 100 samples average is found and the value is assigned to first sample, again for next 100 sample the average value is assigned to 2nd sample. This will not only reduce the number of samples but also remove the fluctuations present.

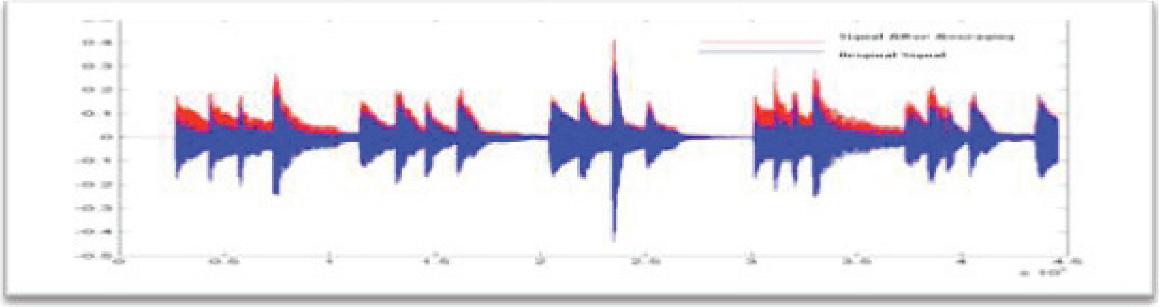

Fig 5. Effect of Averaging

When the decay in the signal is slow, the averaged signal is more densed. But in case of fast decaying the averaged signal represents the envelope of the original signal.

3.1.2 Thresholding

Constant Thresholding : After averaging we need to detect the peaks from the averaged signal. As the name suggests, in constant thresholding one optimum value is decided for which we are able to get maximum number of peaks.

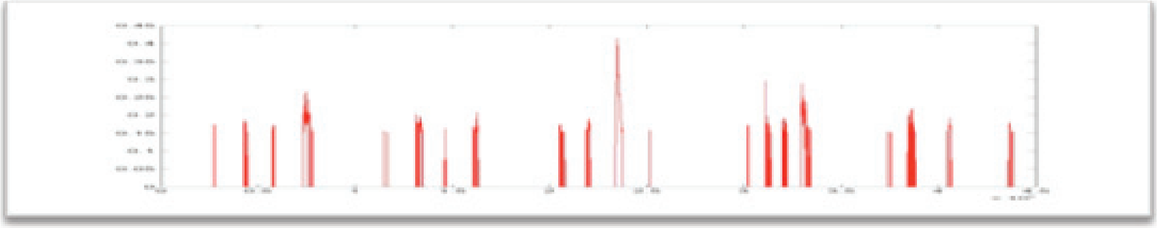

Fig 6. Signal after thresholding

Adaptive Thresholding : There are the possibilities when we take some constant threshold value, for some notes it may be higher than the maximum value of the note and for some notes it is low enough such that two peaks of the note get merged and only one peak (one note) could be detected. To overcome this problem the concept of adaptive thresholding came into picture.

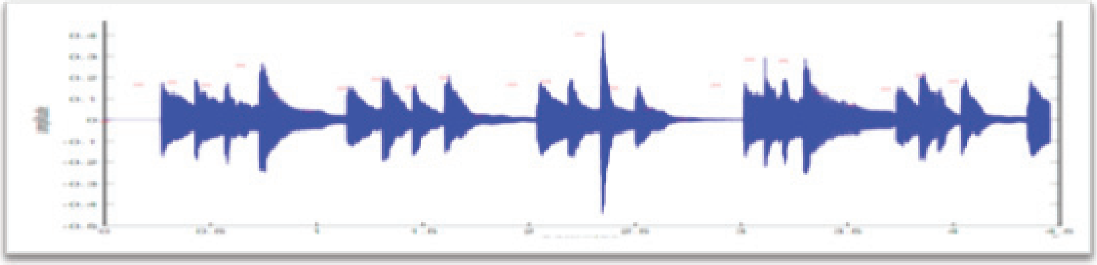

Fig 7. Values of adaptive thresholding

Even in adaptive thresholding one problem is, when there is long silence between two notes, the threshold value for the silence will be very low that supposed to be discarded.

3.1.3 Width Selection for Finding instant

Once the thresholding is done, our aim is to detect the occurences of the peaks from the signal we get after thresholding.

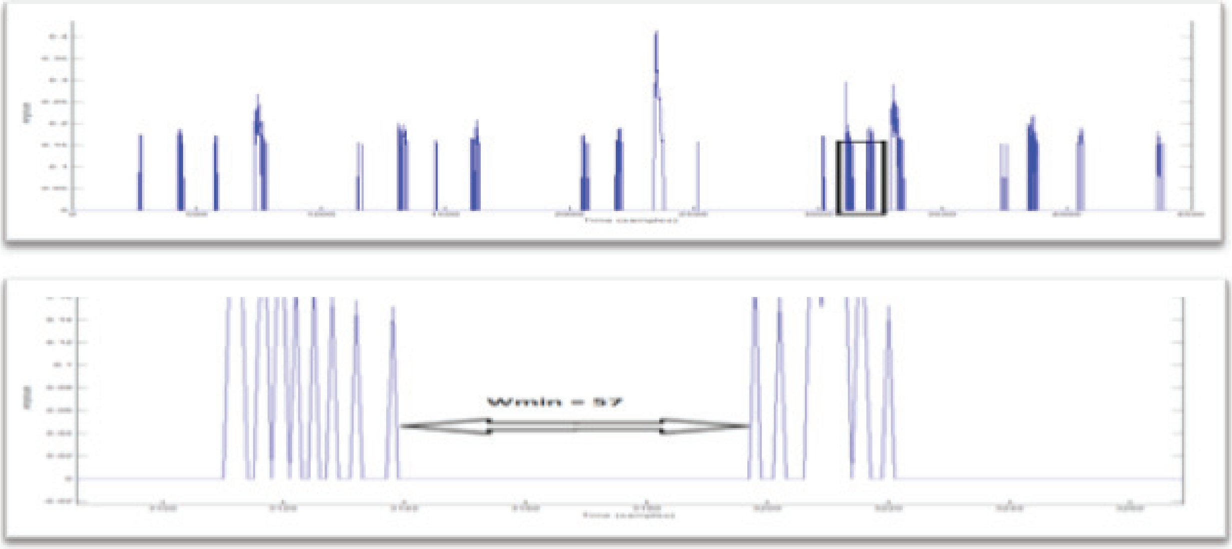

Fig 8. Width selection (a) Signal after thresholding (b) Minimum Width

The width of the window contains all zeroes is selected on the basis of worst case scenario. For our song it is found as 57, so the width was chosen as 50 for safety. The worst case depends on the speed for a given sampling frequency. For slow songs it is more and for faster one it is less. So irrespective of the song, the minimum width was decided as per following equation, $W_{min}=\frac{fs}{(A*20)}$ where, fs = sampling frequency, A = Total number of samples in the original signal that corresponds to one sample in the averaged signal

3.2 I*dentification*

So far we have decided the instants at which the notes are played (key is pressed). The time duration from the first instant to the second instant is the duration of the first note being played, from second to third instant, it's second note being played and so on. Now our aim is to identify these notes based.

3.2.1 Padding Zeroes

For a given note duration, if crop the corresponding instants of the original signal and find the discrete fourier transform, the results are not close enough as required. So we need to pad the zeroes. We can pad the zeroes with different length and different part of the cropped signal and find the DFT that will give the different results. Different part means it can be either only before the cropped version, after it or both the sides with different length. From these different variations the closest results are obtained in the case where the section of zeroes with the same length is padded both the sides of the cropped version. So DFT of the resultant signal is found to determine the notation.

3.2.3 Finding Frequency and Assigning Notation

After padding zeroes, the DFT of the resultant signal is found. Then the corresponding frequency of a particular note is found using equation (1) and the actual note is assigned using table 1.

4. Simulation Results

The experiment was done with different songs like Happy Birthday, Jingle Bell, Twinkle-Twinkle, etc. The results of Happy Birthday are shown below.

Happy Birthday
Total Length (L) = 830063 samples; Time (t) = 26 sec; Sampling Frequency (fs) = 32000 Hz;
Sampling Period (Ts) = 31.25 µs; Total Present Notes (P) =25; Total identifies Notes (D) = 25

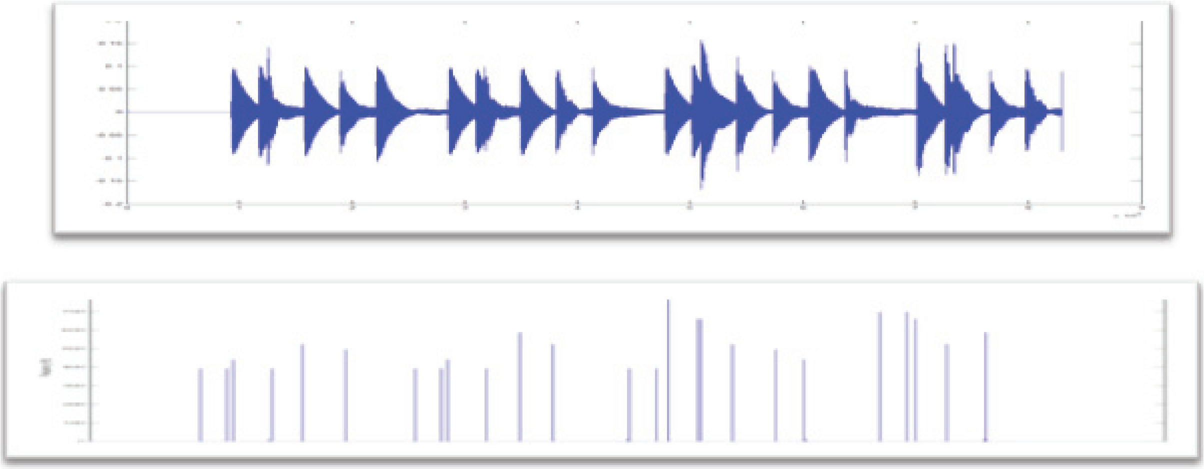

Fig 9. (a) Results Amplitude v/s Time (samples) (b) Frequency (Hz) v/s Time (sec)

Table 2. Results obtained

Sr. No.	Frequency Detected(Hz)	Frequency Assigned(Hz)	Output Notes	time (sec)
1	393.9440624	391.995436	G4	3.0875
2	394.937165	391.995436	G4	3.8375
3	441.6215282	440	A4	3.99375
4	393.7113194	391.995436	G4	5.0875
5	525.7324486	523.2511306	C5	5.9375
6	496.1716113	493.8833013	B4	7.14375
7	393.9087979	391.995436	G4	9.0875
8	395.5372436	391.995436	G4	9.828125
9	441.7132749	440	A4	9.9875
10	393.686712	391.995436	G4	11.09375
11	590.350206	587.3295358	C5	12.00625
12	525.4171797	523.2511306	B4	12.9375
13	393.8059943	391.995436	G4	15.08125
14	394.937165	391.995436	G4	15.84063
15	788.3976356	783.990872	G5	16.17188
16	665.6706126	659.2551138	E5	16.97188
17	662.3520922	659.2551138	E5	17.075
18	525.7559079	523.2511306	C5	17.9375
19	496.0954288	493.8833013	B4	19.14688
20	441.5441926	440	A4	19.94688
21	701.768031	698.4564629	F5	22.07188
22	702.6373901	698.4564629	F5	22.82188
23	662.2568821	659.2551138	E5	23.07188
24	525.6850915	523.2511306	C5	23.9375
25	590.3044482	587.3295358	D5	25.0125

5. Conclusion

In this project, the frequencies of a piano song is detected, corresponding notes are identified with duration. The method used here for note identification is more optimised than previously used methods. By varying the parameters like threshold values and width, we can get the desired results with time duration of each note. Thus project can be treated as an aid tool for learning.

References

1. Prashanth, T. R., & Venugopalan, R. (2011). Note identification in Carnatic Music from Frequency Spectrum. In *Communications and Signal Processing (ICCSP), 2011 International Conference on* (pp. 87-91). IEEE.
2. Kirthika, P., & Chattamvelli, R. (2012,). A review of raga based music classification and music information retrieval (MIR). In *Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on* (pp. 1-5). IEEE.
3. Sridhar, R., & Geetha, T. V. (2009). Raga identification of carnatic music for music information retrieval. *International Journal of recent trends in Engineering*, *1*(1), 571-574.
4. Flexer, A., & Schnitzer, D. (2010). Effects of album and artist filters in audio similarity computed for very large music databases. *Computer Music Journal*, *34*(3), 20-28.
5. Huang, P. S., Chen, S. D., Smaragdis, P., & Hasegawa-Johnson, M. (2012). Singing-voice separation from monaural recordings using robust principal component analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on* (pp. 57-60). IEEE.
6. Levy, M. A. (2005). *Ringermute: An audio data mining toolkit* (Doctoral dissertation, University of Nevada Reno).
7. Chordia, P. (2006). Automatic raag classification of pitchtracked performances using pitch-class and pitch-class dyad distributions. In *Proceedings of International Computer Music Conference*.
8. Shetty, S., & Achary, K. K. (2009). Raga mining of Indian music by extracting arohana-avarohana pattern. *International Journal of Recent Trends in Engineering*, *1*(1), 362-366.