International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

Plagiarism Analysis in Musical Notes

Amutha Prabha N, Abhishek Gudipalli, Vidhyasagar G

***Abstract*: Plagiarism is essentially the copying or stealing of the works of another individual or a group, and presenting them as one's original work. This paper aims to create a plagiarism detection application which analyses two audio files and finds the level of similarity between them, giving a percentage number to aptly define the level of plagiarism in the allegedly copied product. There are a few different techniques which can be used to approach this problem. This work uses Music Information Retrieval to perform analysis and find the similarity in the drum beats and the notes played in the song. A graphical user interface is created to provide ease of use so that everyone – from large corporations to individual end users – can use this application with minimum effort or skill.**

***Index Terms*: Audio, Graphical User Interface, Music, Plagiarism, Audio.**

I. INTRODUCTION

Music is an art form whose medium is silence and sound. The common elements of music are pitch rhythm, dynamics, and the sonic qualities of timbre and texture. Modern music consists of various genres (styles) of music, which have risen due to the different cultures, surroundings, experiences, weathers, developments and many more aspects of life. These may be rock music from the west, or Bollywood music from India, or electronic dance music from all over the world. One thing common in all these types of music is the presence of a musical element – the melody – which is the most important and catchiest part of the song. It may also be referred to as the chorus, which usually consists of musical notes being played repetitively after a certain interval of time. The combination of musical notes together makes a melody.

Another important part of modern music is the presence of a rhythm. Rhythm is an arrangement of sound and silence in time. This gives a tempo to a song which the listeners can catch on to and can relate to the parts of the song. Drum beats are the most common and the most popular way to give a sense of rhythm to a song. These are easily relatable and perceivable by the audience as they are continuously in synchronization with time.

This paper aims to exploit these main features of popular modern music – the presence of a melody and a rhythm. It is used to develop a rudimentary application to analyse and compare two musical pieces and find the similarity between them in order to test for plagiarism, giving a percentage value of similarity. The application compares the audio files on the basis of two important properties in any piece of music – melody (notes) and beats. The application uses a simple FFT analysis to detect the beats of the song and a Cepstrum Analysis to analyse the magnified frequency of each individual note played. This gives a clear analysis of the two pieces of music. The development of a graphical user interface is an important aspect to provide ease of access to our main target audience – small end users.

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author

Amutha Prabha N*, School of Electrical Engineering, VIT University, Vellore, India.
Abhishek Gudipalli, School of Electrical Engineering, VIT University, Vellore, India.
Vidhya Sagar G, School of Electrical Engineering, VIT University, Vellore, India.

II. DESCRIPTION

An application is required to be developed to provide a numeric value to the amount of plagiarism in the copied piece of music. A graphical user interface is developed in MATLAB using the Guide tool for individual users to select the audio files they want to compare, an estimate of the power spectrum of the audio files is displayed and the frequencies of the notes played along with the time is displayed for easier comparison.There are several methods which could be used to analyze an audio file and retrieve information from it, one common method being audio fingerprinting and similarity detection [4]. But this method is slow to give results and may not be as accurate. It requires the creation of databases to compare the fingerprint of each audio file after a query and then give the result. This method also doesn't give any other information regarding the audio file which may be useful in the application of real time music.Auto-correlation method is used to find the BPM (beats-per-minute) of the audio files and the Cepstrum analysis to find the magnified frequencies of each note being played in the musical pieces. This gives us a visual representation of the sound bring played which makes it easily comparable.

A. Autocorrelation for BPM Detection

In signal processing cross-correlation is a method to estimate how similar two signals are. It is defined (in the discrete case) as –

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f^*[m]\, g[n+m] \qquad (1)$$

f* is the complex conjugate of 'f'. Auto-correlation therefore is a cross-correlation of a signal with itself at a given lag or delay. By carrying out the auto-correlation of a signal, one can identify at what lag the signal is most similar to itself. For most periodic music with a strong beat, the highest correlation will occur on the beat and one can therefore perform a quick calculation to convert the lag index (number of samples delayed) of the greatest correlation to a BPM value.

In our approach, the input audio signal is separated into different frequency bands using discrete wavelet transform (DWT). Mainly, each band corresponds to the energy in an octave. The envelope of each band is calculated using full wave rectification, low pass filtering and normalization. The purpose of the envelope extraction step is to retain events in that particular octave without being affected by particular pitches. The normalized envelopes are added with the effect of enhancing common periodicities with the same phase. The sum of the normalized envelopes is then processed through an autocorrelation function to detect the dominant periodicities of the signal [3] as in fig.1.

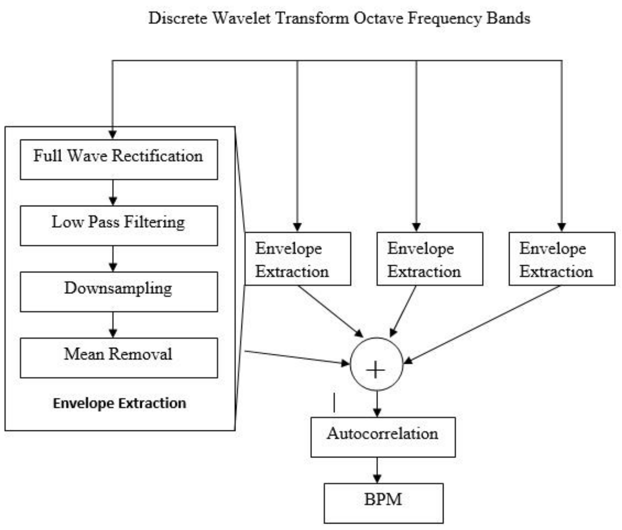

Fig.1. BPM Detection Algorithm

B. Pitch Tracking using Cepstrum Analysis

Pitch Tracking using Cepstrum Analysis:
All Pitch tracking is a method for estimating the pitch or fundamental frequency of a quasiperiodic or oscillating signal, usually a speech signal or a musical note or tone. The tracking can be done in time domain as well as frequency domain. This method gives the information of pitch level of a signal. Hence it can be used to extract information in various fields, such as music information retrieval, phonetics, speech coding, etc.
Cepstrum Analysis is a method to track the pitch of a sound or a musical signal. It gives the frequency of notes played in an audio file. Fig.2. represents the various steps involved in Cepstrum Analysis are:

- The audio signal is displayed in the Hamming Window.
- This signal then undergoes various processes such as FFT, logarithm and Inverse FFT.
- First, the FFT coverts signal from time domain to frequency domain.
- Then, the logarithm function removes the sharp peaks produced by FFT.
- Finally, the inverse FFT generates periodic pulse, determining the frequency levels of the audio signal.

The power Cepstrum of a signal = $|\mathcal{F}^{-1}\{\log(|\mathcal{F}\{f(t)\}|^2)\}|^2$
Where f(t) is the function of the audio signal in time domain.

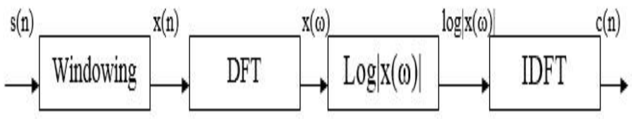

Fig.2. Steps in Cepstrum Analysis

The Cepstrum Analysis of a C-4 major scale played on a piano yields the following results as in fig.3

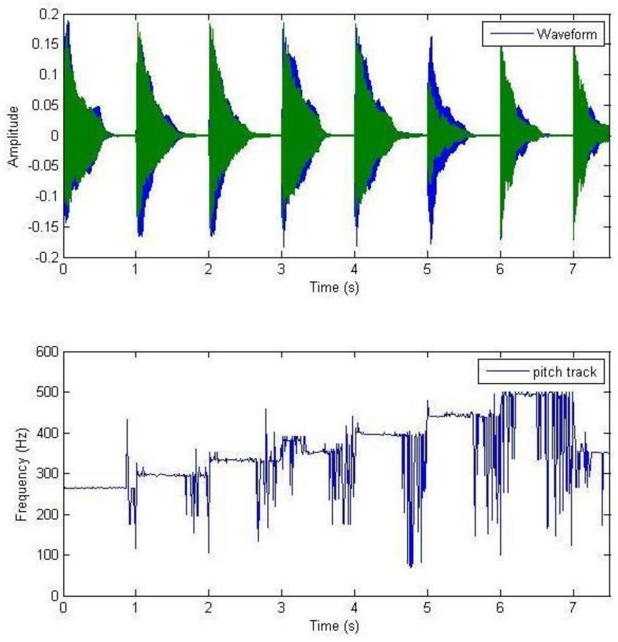

Fig.3. Cepstrum Analysis of C-4 major scale

The results are accurate as the first note played – C-4 major has the frequency of 261 hertz, the frequency of the second note played – D-4 is approximately 293 hertz and so on. This proves that the results obtained are accurate. To test the functionality of the algorithm, a more complex scale was used with a non-rhythmic timing pattern. The results obtained are displayed on Axes5 and the Percentage Similarity is displayed in Text Editor in fig.4.

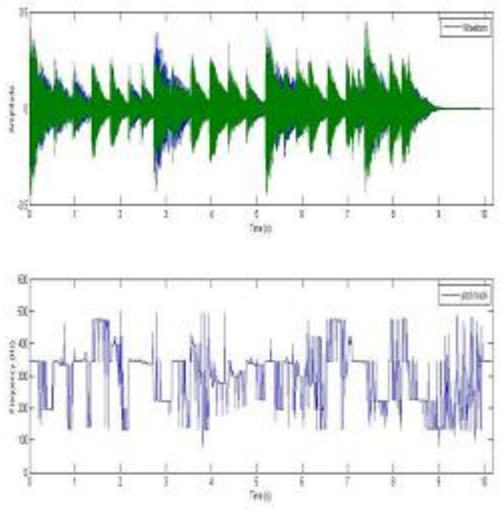

Fig.4. Cepstrum Analysis of Complex Scale in C

III. APPLICATION DESIGN

A user-friendly interface is required to demonstrate the relation between two audio (.wav) files. It should extract the information from two audio files and compare this information to display similarity between the files, in visual as well as data form. MATLAB GUI has features suited to such requirements.

This GUI has options to select files according to user's requirements. It displays plots to demonstrate the information obtained from the audio files. Finally, it displays the value of percentage similarity between the files by comparing the individual data obtained from them.

A. GUI Design Layout

- MATLAB "guide" is used in Command Window to design the layout of GUI.
- Blank GUI is selected. A blank GUI editor pops up.
- Buttons "Sound File 1" and "Sound File 2" are used to open browser window to select the sound files which are to be compared.
- "Amplitude Vs Time" response for the sound files selected is displayed on Axes1 and Axes2 respectively.
- "BPM" buttons are used to track Beats per Minute of sound files with percussion. The values are displayed in the Text editors respectively.
- The "Compare" button is used to do comparison between the two sound files. "Frequency Vs Time" graphs of the files are displayed on Axes3 and Axes4. The similarity plot is displayed on Axes5. And the Percentage Similarity is displayed in Text Editor as shown in fig.5

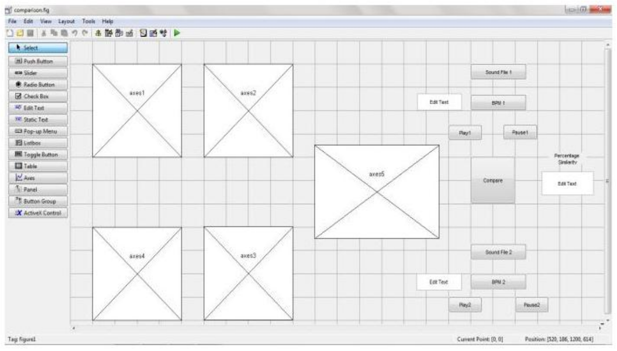

Fig.5. Application GUI Framework

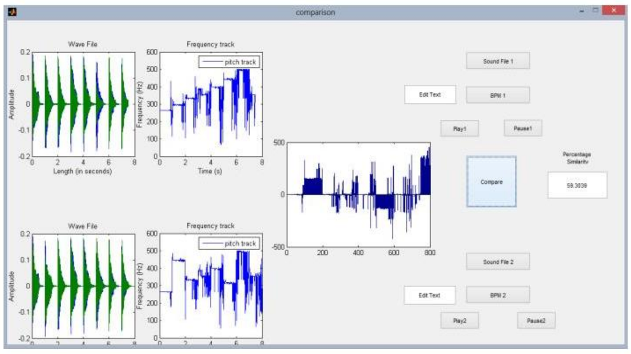

Fig.6. Working GUI

B. Note Recognition

The note recognition algorithm used in the application to identify individual notes being played in the scale is explained in reference to the GUI developed and represented in fig.6

- Number of elements in the final matrix (bar graph elements), that is obtained by taking the difference of the sound files is stored.
- Two variables are set as the starting and end of a note.
- For each note, the average frequency is checked and this value is sent to the function "note".
- Here, the note's scale is checked by comparing its frequency to actual frequency of different notes.
  - The name of the note (for e.g. - C, D, etc.) is returned and displayed in the command window.

IV. RESULTS

The application gave a good percentage of similarity when only few notes of the C-4 Major Scale were changed and compared with the original. The different notes being played in the audio file were also recognized with accuracy and the difference between the notes played between two audio files was recognized and a percentage value of this difference was calculated. The notes for the first signal were: C, D, E, E, F, and G. The notes for the second signal were: C, D, E, F, D and G. Preliminary test to check working is to input the same audio file to get a 100% similarity between the two as the FFT and Cepstrum algorithms should give the same output..

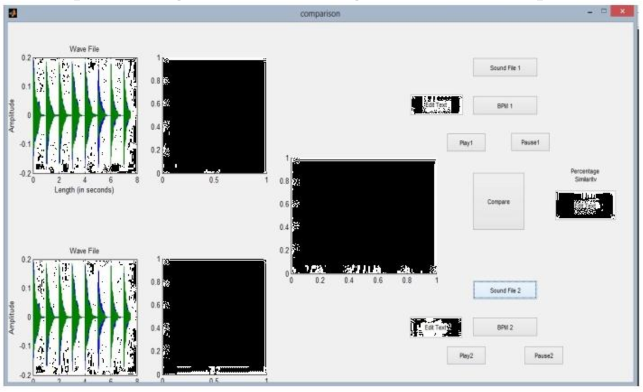

Fig.7. Same Audio File in both Inputs

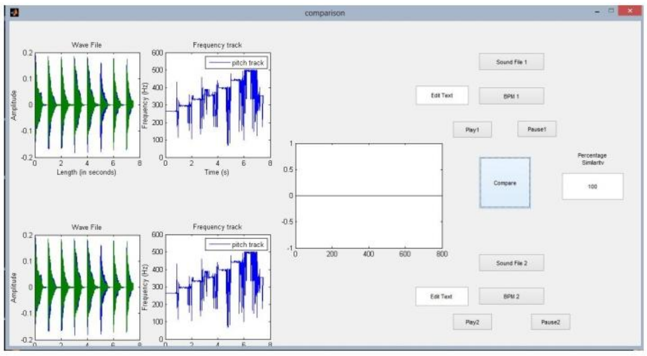

Fig.8. 100% Similarity for given inputs

Fig.7 indicates the pitch track of the files are identical, hence they both cancel each other to give a 0 value throughout the time domain. Hence, same audio gives a 100% similarity. The obtained frequency track of the audio file also gives the frequency of each note played in the audio file. These notes are obtained by comparing the frequencies of notes in the signal to the real frequencies of various notes as in fig.8.

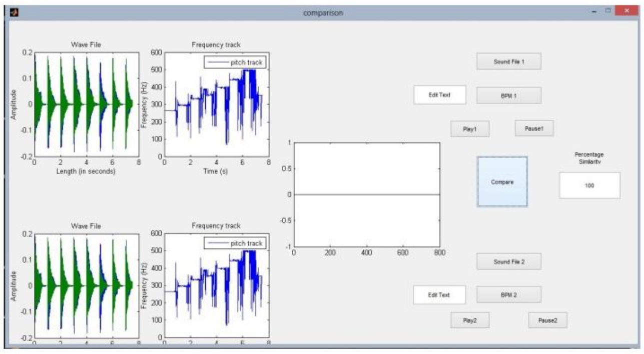

Fig.9. Power Spectrums of two dissimilar inputs

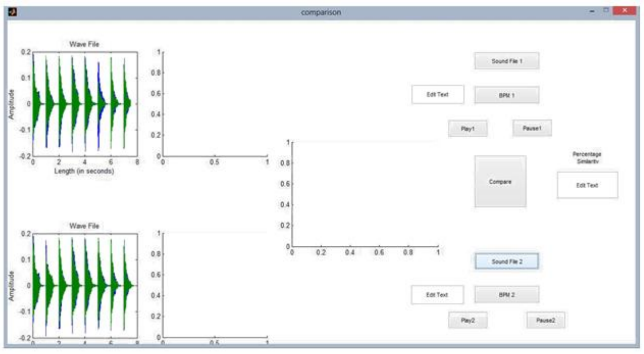

Fig.10. 59.9% similarity for given inputs

Fig.9 represents an array of elements after comparing the two files is obtained. The elements whose value is close to 0 are the parts of music that is similar. This matrix gives a percentage similarity of 59% as obtained in fig.10. The obtained frequency track of the audio file gives the frequency of each note played in both scales.
It can directly be observed that 2 of the 6 notes are different. Theoretically, 4 out of 6 notes are similar. Hence percentage similarity should be approximately 4/6 i.e. 66.67 %.The GUI considers the noise and timings of the notes. Hence, the value obtained after considering all the factors is 59%. Also, 100% similarity was obtained when same file was compared to itself.

V. CONCLUSION

This application will protect an individual's copyrights and will encourage more creativity in music industry. It will give more recognition to real talent and deter content theft. It can improve upon the features and be modified to make it more user-friendly and track similar sounds whose scale (pitch) is changed. Also, melody extraction can be added to decompose a song into individual instruments and compare them individually.

REFERENCES

1. C. Dittmar, K. Hildebrand, D. Gaertner, M. Winges, F. Muller, P. Aichroth, "Audio Forensics Meets Music Information Retrieval - A Toolbox for Inspection of Music Plagiarism",EURASIP European Signal Processing Conference (EUSIPCO'12), Bucarest, Romania, pp.- 1249-1253, August 2012
2. Jay K. Patel, "Musical Notes Identification using Digital Signal Processing", 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), pp.-876-884, July 2015
3. George Tzanetakis, "Tempo Extraction using Beat Histograms", University of Victoria, Computer Science Department, March 2005
4. Milind Bhattacharya, Sweekar Bandkar, Amit Badala, "Music Analyzer and Plagiarism", Vidyalankar Institute of Technology, Mumbai University, May 2014.
5. Juwan Lee, Sanghun Park, Seokhwan Jo and Chang D. Yoo, " Music Plagiarism Detection System", Dept. of Electrical Engineering, KAIST, Daejeon, Korea, June 2011.
6. Daniel P. W. Ellis, "Beat Tracking by Dynamic Programming", Columbia University, July 2007

AUTHORS PROFILE

N.Amutha Prabha completed her undergraduate in engineering (Electronics and Communication Engineering) and Masters in Applied Electronics. Received her Doctorate in Wireless communication from Anna University. She published more than 40 publications in an International / National Journals and conferences. Her area of interest includes Wireless/Mobile Communication, Signal / Image processing and Biomedical engineering.

Dr. Abhishek Gudipalli has joined to school of electrical engineering, VIT University, Vellore on 05th December 2011 as an Assistant Professor. He has done his B.Tech (Electronics and Communication Engineering) from PBR Visvodaya institute of technology and science, kavali in 2005, M.Tech (Sensor Systems and Technology) from VIT University, Vellore in 2007 and Ph.D. (Development and analysis of hybrid edge detection algorithms for color models) from Sri Venkateswara University, Tirupati in 2016. He has done master thesis under student exchange program at University of applied sciences, Karlsruhe, Germany during the year 2006-07. He is an author of about 30 papers published in journals and as well as national and international conferences.He is a life member of the IEEE and ISTE. His research interests include Image Processing, Signal Processing, Pattern Recognition, Machine Learning, Sensor systems, wireless sensor networks, virtual instrumentation..

G Vidhya Sagar, completed his undergraduate in Electrical and Electronics Engineering and received the M.tech degree in VLSI systems from National Institute of Technology Trichy(NITT), Trichy, Tamil Nadu, India, and currently working towards the Ph.D. degree at VIT university Vellore, Tamil Nadu, India. Since July 2010, he has been with VIT University, Vellore, Tamil Nadu, India, working as Assistant professor. His research topic is Hybrid FinFET device modelling in Analog and RF applications.