Chord Identification Using Pitch Class Profile Method with Fast Fourier Transform Feature Extraction

Kurnia Muludi[1], Aristoteles[2], Abe Frank SFB Loupatty[3]

[1]Computer Science Department, The University of Lampung, Bandarlampung 35145, Lampung, Indonesia

[2] Computer Science Department, The University of Lampung, Bandarlampung 35145, Lampung, Indonesia

[3] Computer Science Department, The University of Lampung, Bandarlampung 35145, Lampung, Indonesia

Abstract

Recognizing a chord of a music is not an easy task to do, but it can be done easier using digital signal processing in computer system. This research implements pitch class profile (PCP) method in recognizing chord from audio signal. The audio signals are converted from time domain into frequency domain using Fast Fourier Transform (FFT). In this research, signals are processed using various frame blocking width in order to find out the most optimal width in frame blocking using Hamming window. This research implements chord recognition method in Python programming to process guitar chord recording. The purpose of this research is to find out the accuracy of chord recognition using PCP method.

Keywords: *Chord; Guitar; Pitch Class Profile; Fast Fourier Transform.*

1. Introduction

It is understood that chord is the most basic guitar playing. Chord playing is also the basic element in a song. To recognize chord is not an easy task to do, because chord doesn't consist of single pitch. Moreover, a person with good musical ability is not surely capable to do it. A person with no musical experience needs long time effort to recognize a single pitch of music. But musical chord of audio signal can be recognized using signal processing technique with computer. Computer can recognize tone of a audio signal by measuring the fundamental frequency of a audio wave and match the frequency with the reference frequency. With signal processing, computer can recognize not only pitch but also other parameters of a signal those can be used to recognize a chord.

Processing an audio signal takes a process called digital signal processing (DSP). DSP is a process that analyze input and output of a signal to know the certain physical system characteristic of a signal. Analyzing and synthesizing signal can be done easier at the frequency domain, because the determinant factor of signal is the frequency [12]. Therefore, the earliest stage that is necessary in DSP is feature extraction to convert signal from time domain into frequency domain. After that conversion, processing the signal can be done easier.

In this research, a pitch identification system of audio signal has built to recognize the chord as the output. System was developed using fast fourier transform (FFT) as feature extraction technique and pitch class profile (PCP) as a method to identify the chord. This research shows the accuracy of chord recognition using PCP method.

2. Research Stages

The stages those conducted in this research are; literature review, data retrieval (chord recording), signal sampling, signal preprocessing, pitch class profile mapping, chord determination and testing.

2.1 Data Recording

Guitar chord are recorded using USB Guitarlink as interface between computer and guitar. The recorded files are processed with Steinberg Nuendo software. As many as 432 chords are recorded during 2 seconds to be used as sample in this research. Using Steinberg Nuendo, 432 files with *.wav extension, 44100 KHz sampling frequency, 16 bit rate and mono channel are generated.

2.2 Building The Database

In this research, the database is composed of the values of PCP vector those are expected from the tested chord. Each pitch classes in PCP vector are contain value 1 or 0, value 1 represents the pitch contained in chord, whereas value 0 means the otherwise. The forming of C Mayor chord PCP vector can be seen in the Figure 1.

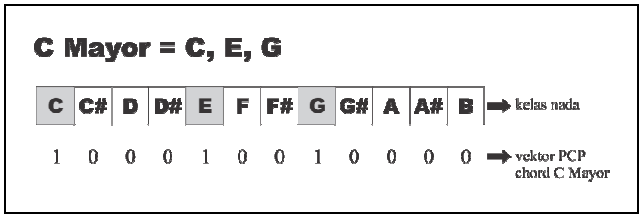

Fig. 1 The forming of C Mayor chord.

2.3 PCP Implementation on Python Programming

In order to implements PCP technique in recognizing guitar chord, an application program is built with Python programming. With this application, every sample can be tested using PCP technique. The frame width of frame blocking process can be specified in this application. The application that built can be seen in the Figure 2.

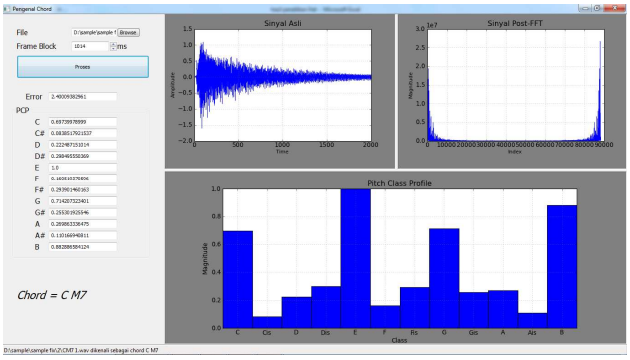

Fig. 2 The chord identificator application.

2.4 Frame Blocking and Windowing

In application, frame blocking process is implemented with overlap at 50% of its frame width. In this stage, each frames are multiplied by Hamming window. The size of Hamming window is as wide as frame width in frame blocking process. This stage is realized by Python program code below.

```
#BUAT WINDOW HAMMING
overlap = lebarFrame / 2
window = []
for i in range(lebarFrame):
    window.append(0.54 - (0.46 * cos((2*pi*i)/lebarFrame-1)))

#FRAME BLOCKING
j = 0
sinyalFB = []
for i in range(len(sinyal)):
    sinyalFB.append(sinyal[i] * window[j])
    j += 1
    if (j > (lebarFrame-(overlap/2))):
        j = overlap / 2
```

Fig. 3 Frame blocking and windowing in Python code.

2.5 FFT and Peak Detection

After the previous stage, conversion process from time domain to frequency domain is conducted using fast fourier transform. FFT function used is function that is in the Matplotlib library of Python Programming. Frequency peaks obtained from FFT process then used in PCP stage to create the PCP vector. Figure 4 is the pieces of the program code that implements the FFT and frequency peak detection stage.

```
#TRANSFORMASI KE DOMAIN FREKUENSI DENGAN FFT
sinyalFFT = fft(sinyalFB)
sinyalFFT = abs(sinyalFFT)

#DETEKSI PUNCAK DAN FREKUENSI
frekuensi = []
magnitude = []
for i in range(len(sinyalFFT)/2):
    if (sinyalFFT[i] > 1000000):
        if (sinyalFFT[i]>sinyalFFT[i-1]) & (sinyalFFT[i]>sinyalFFT[i+1]):
            frekuensi.append((i+1)*44100/len(sinyalFFT))
            magnitude.append(sinyalFFT[i])
```

Fig. 4 FFT and peak detection in Python program code.

2.6 PCP

After the preprocessing stages, frequency peak values obtained are grouped into defined pitch classes to create the PCP vector. The pieces of program code that implements PCP method to create PCP vector can be seen in Figure 5.

```
#PCP
nfrek = [61.87, 66, 68.75, 74.25, 77.34, 82.5, 88,
         91.67, 99, 103.1, 110, 117.3, 123.75, 132]
pcp = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
for i in range(len(frekuensi)):
    for oktaf in range(8):
        for nf in range(1,13):
            if (frekuensi[i]>(2**oktaf*((nfrek[nf-1]+nfrek[nf])/2)))&
               ((2**oktaf*((nfrek[nf]+nfrek[nf+1])/2))>frekuensi[i]):
                pcp[nf-1] += magnitude[i]
```

Fig. 5 Pitch class profile in Python program code.

3. Test Result and Discussion

In this research 432 recorded chord data are tested into the program in order to find out the amount of data that succesfully recognized and the amount that is fail to be recognize as the expected chord. Each data are tested using various frame blocking width, such; 15, 20, 25, 35, 40, 50, 200, 661, 850, 1014, 1313, and 1323 data bit.

3.1 The Effect of Frame Blocking Width on Accuracy

Based on testing that has been done, the highest accuracy is obtained in frame blocking using 1014 data bit frame width. In testing using 1014 bit data frame width 318

chord samples are succesfully recognized, whereas 114 samples are not. Accuracy obtained in testing using 1014 data bit frame width is 73,61%. The accuracy obtained in testing using various frame blocking width can be seen in Figure 6.

Fig. 6 Chord accuracy in various frame blocking width.

3.2 The Effect of Chord Type on Accuracy

Based on experiment, various accuracy is resulted from testing in various type of chord. From six chord type those are tested, Sustained 4 chord type generates the highest accuracy value. While, Dominant 7 chord type generates the lowest accuracy value. Figure 7 below shows the accuracy obtained from testing in each type of chord using various frame blocking width.

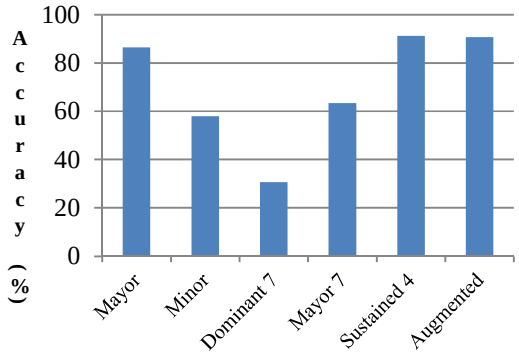

Fig. 7 Chord accuracy in chord type.

To finds out the consistency of this result, the result data are grouped by frame blocking width. The result of all group are identical, the lowest accuracy is always generated by Dominant 7 chord type. While Mayor, Sustained, and Augmented chord type always generates accuracy above 79%.

3.3 The Effect of Fret Position on Accuracy

In research the effect of fret position on chord recognition accuracy. Tested samples are grouped by the lowest fret guitar position when it played. The result of each group then averaged as can be seen in Figure 8.

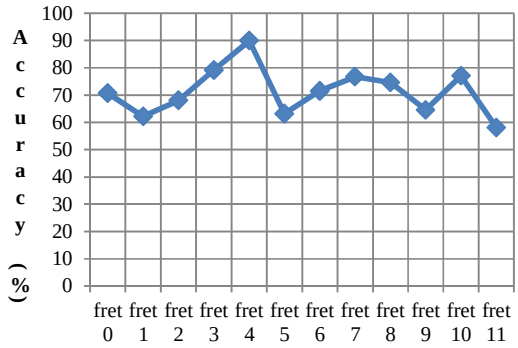

Fig. 8 Chord accuracy in fret position.

Based on the figure, the standard deviation generated from the result is 0,0887. Because of that, it is conclude that fret position doesn't affect the accuracy level of chord recognition

3.4 The Effect of Fingering on Accuracy

Testing on effect of chord fingering (fingers position that play chord on guitar fretboard) is conducted. Therefore, data are grouped into 3 chord fingering group; open chord, barre chord 1, and barre chord 2.

1. Open Chord is the group of chord that contain at least 1 open string rung.
2. Barre Chord 1 is the group of chord that has root on 6th string start at fret 1.
3. Barre Chord 2 is the group of chord that has root on 5th string start at fret 1.

Figure 9 is the result of testing on effect of chord fingering. Barre chord 2 generates the highest accuracy, as big as 76,56%. Open chord generates 68,25% accuracy, while barre chord 1 has the lowest accuracy, as big as 63,87%.

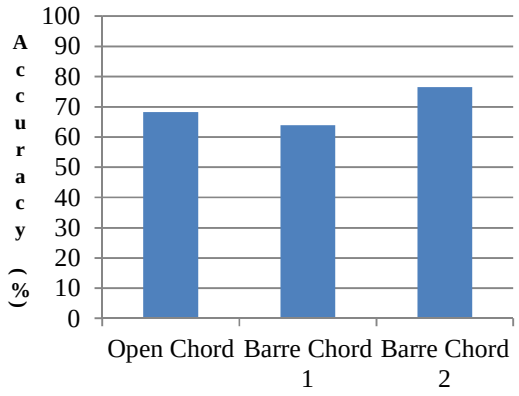

Fig. 9 Chord accuracy in fingering.

3.4.1 The Effect of Fingering on Mayor Chord Accuracy

In Mayor chord type, open chord has the lowest accuracy. While barre chord 1 and 2 generates accuracy level above 80%. Barre chord 1 has the highest accuracy, as big as 95,99%.

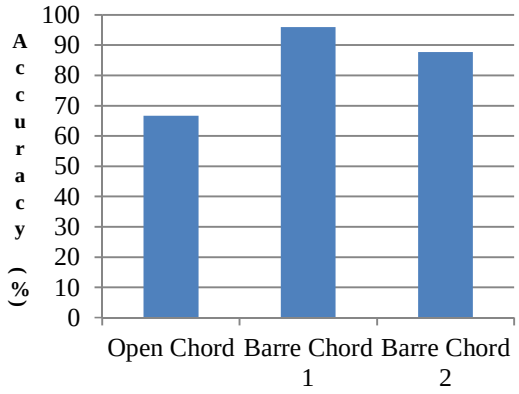

Fig. 10 Chord accuracy in fingering of mayor chord.

3.4.2 The Effect of Fingering on Minor Chord Accuracy

In Minor chord type, lowest accuracy is generated by barre chord 1, with accuracy level less than 10%. In barre chord 1, all testing on F minor and C minor are fail to recognize as expected chord. While all chord in barre chord 2 generates expected chord, that generates 100% accuracy level.

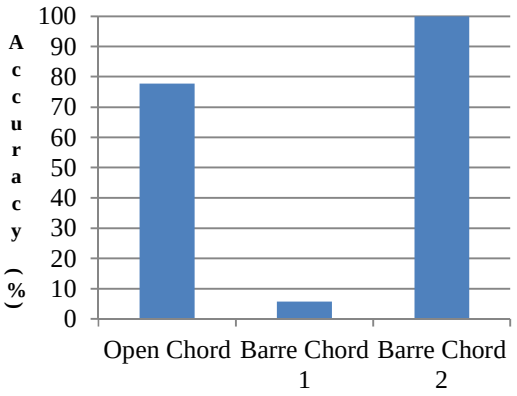

Fig. 11 Chord accuracy in fingering of minor chord.

3.4.3 The Effect of Fingering on Dominant 7 Chord Accuracy

In the testing on Dominant 7 chord accuracy obtained from each group are less than 53%. The highest accuracy is generated by open chord, as big as 52,78%. While the lowest accuracy is obtained from barre chord 1, with 20% accuracy level. In this testing, all testing on F7, G7, and A#7 barre chord 1 generate all fail recognition.

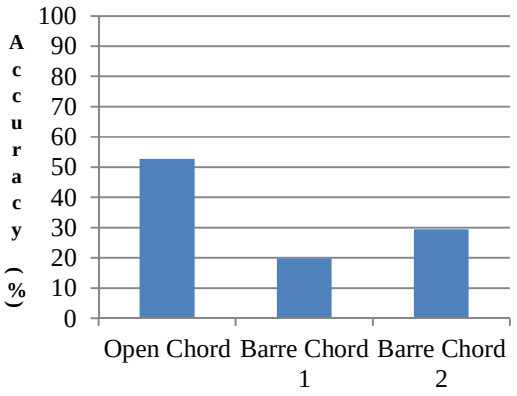

Fig. 12 Chord accuracy in fingering of dominant 7 chord.

3.4.4 The Effect of Fingering on Mayor 7 Chord Accuracy

In the testing on Mayor 7 chord type all group generates accuracy less than 70%. Open chord group obtain 49,07% accuracy level which is the lowest. The highest accuracy is generated by barre chord 1, while barre chord 2 obtain 63,38% accuracy level.

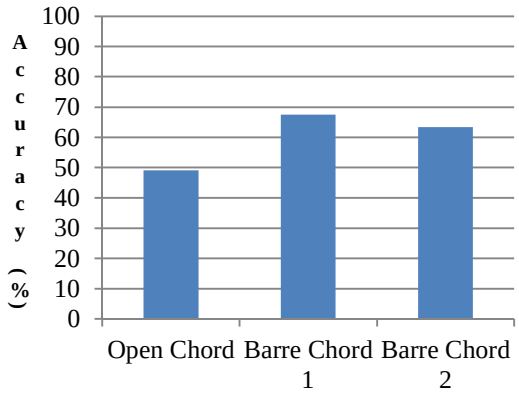

Fig. 13 Chord accuracy in fingering of mayor 7 chord.

3.4.5 The Effect of Fingering on Sustained 4 Chord Accuracy

In the testing on Sustained 4 chord type, all group generate accuracy above 80%. Barre chord 1 group obtain 96% accuracu level which is the highest. The lowest accuracy is generated by open chord group with 83% accuracy level.

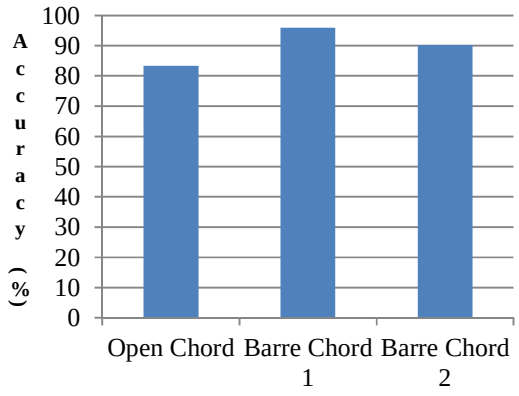

Fig. 14 Chord accuracy in fingering of sustained 4 chord.

3.4.6 The Effect of Fingering on Augmented Chord Accuracy

In this testing, barre chord 1 group generates the highest accuracy level at 98%. While open chord group obtain the lowest accuracy with 79,86% accuracy level.

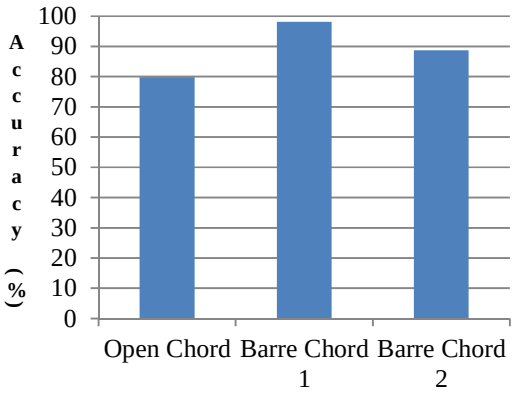

Fig. 15 Chord accuracy in fingering of augmented chord.

4. Conclusions

Based on this research, these following things can be concluded:

1. This research succesfully implements PCP method in recognizing guitar chords with 70,06% accuracy level.
2. Highest accuracy is obtain in the testing with 1014 data bit frame blocking width.
3. The success of PCP method in recognizing chord is affected by frame blocking width, chord fingering, and the chord type.
4. In this research, each chord type generates different average accuracy. Dominant 7 chord type generates the smallest average accuracy.
5. In Dominant 7 chord type, barre chord 1 fingering group generates the lowest average accuracy. This is because some strings in Dominant 7 fingering don't ring loud enough. This thing make the value of pitch class in PCP vector that is generated isn't big enough.

References

[1] Agustini Ketut. 2008. *Perbandingan Metode Transformasi Wavelet Sebagai Praproses Pada Sistem Identifikasi Pembicara*. Bogor: Sekolah Pasca Sarjana Institut Pertanian Bogor.

[2] Cabral, G, Briot J-P, dan Pachet F. *Impact of Distance in Pitch Class Profile Computation.* Paris: Laboratoire d'Informatique de Paris, Universite Pierre et Marie Currie.

[3] Fadlisyah, Bustami, Ikhwanus M. 2013. *Pengolahan Suara*. Yogyakarta: Graha Ilmu

[4] Fruandta Ade. 2011. *Identifikasi Campuran Nada pada Suara Piano Menggunakan Codebook.* Departemen Ilmu Komputer FMIPA IPB

[5] Gaffar Imam, Hidayatno Achmad, dan Zahra Ajub Ajulian. 2012. *Aplikasi Pengkonversi Nada-nada Instrumen Tunggal Menjadi Chord Menggunakan Metode Pitch Class Profile.* Jurusan Teknik Elektro, Universitas Diponegoro Semarang.

[6] Hendro. 2007. Memperindah Permainan Keyboard Dengan Menggunakan Jembatan Akor. Jakarta: Kawan Pustaka.

[7] Huda Miftahul, Basuki Dwi Kurnia, Akbar Fandy, dan Permana Febrianzah Junaidy. 2009. *Konversi Nada-Nada Akustik Menjadi Chord Menggunakan Pitch Class Profile.* Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember

[8] Kushidayati Milla Fitriani, Huda Miftahul, Setiawardhana. 2010. *Pembuatan Database Transkrip Akord Instrumen Tunggal Menggunakan Metode Enhanced Pitch Class Profile (EPCP).* Jurusan Telekomunikasi Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember.

[9] Marchand, S. 2001. *An Efficient Pitch-Tracking Algorithm Using a Combination of Fourier Transforms*. Limerick: Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01).

[10] Mustofa Ali. 2007. *Sistem Pengenalan Penutur dengan Metode Mel-frequency Wrapping* . Jurusan Teknik Elektro, Universitas Brawijaya.

[11] Putra Darma, Resmawan Adi. 2011. *Verifikasi biometrika suara menggunakan metode MFCC dan DTW*. Fakultas Teknik Universitas Udayana.

[12] Riyanto Sugeng, Purwanto Agus, dan Supardi. 2009. *Algoritma Fast Fourier Transform (FFT) Decimation In Time (DIT) dengan Resolusi 1/10 Hertz*. FMIPA Universitas Negeri Yogyakarta.

[13] Smith, S W. 1998. *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego: California Technical Publishing.

[14] Sumoro Hadi. 2007. *Frekuensi Dasar, Harmonics dan Overtones.*

[15] Timotius Ivanna K, Prayogo Adhi. 2010. *Sistem Pengenalan Chord pada File Musik Digital dengan Menggunakan Pitch Class Profiles dan Hidden Markov Model*. Salatiga: Fakultas Teknik Elektronika dan Komputer, Universitas Kristen Satya Wacana.

[16] Wisnudisastra Elghar. 2009. *Pengenalan Chord pada Alat Musik Gitar Menggunakan Codebook Dengan Teknik Ekstraksi Ciri MFCC.* Departemen Ilmu Komputer FMIPA IPB