NOTE DETECTION FOR SAZ WITH HARMONIC PRODUCT SPECTRUM METHOD

Zafer Ozcan, Bambang Hidayat, Unang Sunarya
Bachelor Degree Telecommunication Engineering, School of Electronic Engineering, Telkom University
zaferoz91@gmail.com unangsunarya@telkomuniversity.ac.id

Abstract - This research is about automatic note detection for Turkish instrument "Saz". The Harmonic Product Spectrum (HPS) method is used for detection. The reason of doing this research is that so many applications are created for usage of music industry. Most of the music that we listen today is usually created in digital studios. However there are only few research on Turkish traditional instruments. Listening to a melody and extracting its notes is not possible for everyone. Therefore the musician publishes the notes on a paper then others can play the same melody. Today, even if we don't have the notes written on a paper we can extract the notes with a software. This is possible for many instruments, such as guitar, piano, clarinet etc. however there is no application created to detect notes of Saz yet. Therefore this research is important for increasing the usage of Saz by Musicians.

In this research, Harmonic Product Spectrum (HPS) method is used as feature extraction method. HPS applies down sampling to the input data in spectrum format then multiplies the each result of the down sampled input. Saz Notes are recorded in Signal Laboratory, as Wav file. A Matlad based algorithm is used to simulate the system. Notes are fed to the system offline. Firstly. The input is preprocessed and converted to feasible form for HPS. Secondly, preprocessed data is fed to the HPS. HPS extracts the feature of the note. The feature is consist of two frequency value. Finally the features are compared with the training data with k-NN method to identify the note.

In experiments we achieved 97% accuracy for noiseless notes and 89% accuracy for noisy notes. The software is able to extract the feature around 0.01 second. Overall computation time is around 0.0603 second. In a very similar research, done by Paul M. Brossier, the accuracy is 96% and the computation time of their software is 0.3 second. In another research, done by Chris Duxbury, the accuracy for onset detection was 92%. The common accuracy level for detection din researches about musical instruments varies from 92% to 98%. There are some rare researches achieved 100% accuracy level. Therefore the accuracy level of our research can be labeled as very satisfying..

Keywords: Saz, Harmonic Product Spectrum, Pitch, Note Detection

I. INTRODUCTION

Technology has developed a lot in music industry. Today a musician can tune the guitar or piano with a simple android application. A musician can play a random melody with guitar and momentarily can observe the notes he/she plays. Or a musician can create a song with multiple instruments without even using real instruments. Because so many researches are done for musical audios and so many software and applications are developed years ago. But still researches on musical objects are of importance. Turkish Folk instruments also intensively used in musical applications. One of the most important Turkish folk instrument is Saz. Because in almost all Turkish folk music Saz is played. Therefore we chose Saz for the research. In this work we aim to create an algorithm which can identify notes played with Saz.

Various approaches to note detection have been proposed. For example Simon opted for a combined time and frequency domain approach, with the system being trained on part of the data for optimal detection for piano. His method was quite successful in accuracy of detection. However computation cost of his algorithm was high as well. Another research similar to ours is "Fast labelling of notes in music signals", proposed by Paul M. Brossier, Paul also uses fundamental frequency to classify the notes. He has reached 96% accuracy level with 0.3 second computation time. Our algorithm and method offers slightly higher accuracy, which is 97%, and remarkable speed of computation with 0.06 second.

II. BASIC THHEORY

A. Saz

Saz is the most commonly used string folk instrument in Turkey. It takes different names according to the regions and according to its size such as Baglama, Divan Sazi, Bozuk, Çögür, Kopuz Irizva, Cura and Tambura [1]. An example of Saz is depicted in Fig. 1.

Figure 1: An example of Saz

B. Harmonic Product Spectrum

This algorithm estimates the pitch as the frequency that maximizes the product of the spectrum at harmonics of that frequency, as

$$p = argmax \prod_{k=1}^{n} |X(kf)| \tag{1}$$

where *X* is the estimated spectrum of the signal, *n* is the number of harmonics to be used (typically between 5 and 11), and *p* is the estimated pitch. The purpose of limiting the number of harmonics to *n* is to reduce the computational cost, but there is no empirical reason behind this limit; therefore we are able to choose the number of harmonics, but we must mind that computation time increases with respect to the amount of harmonics. Since the logarithm is an increasing function, an equivalent approach is to estimate the pitch as the frequency that maximizes the logarithm of the product of the spectrum at harmonics of that frequency [2].

Here we propose how we used the harmonic product spectrum method to extract the fundamental frequency by multiplying the several times down sampled input signal.

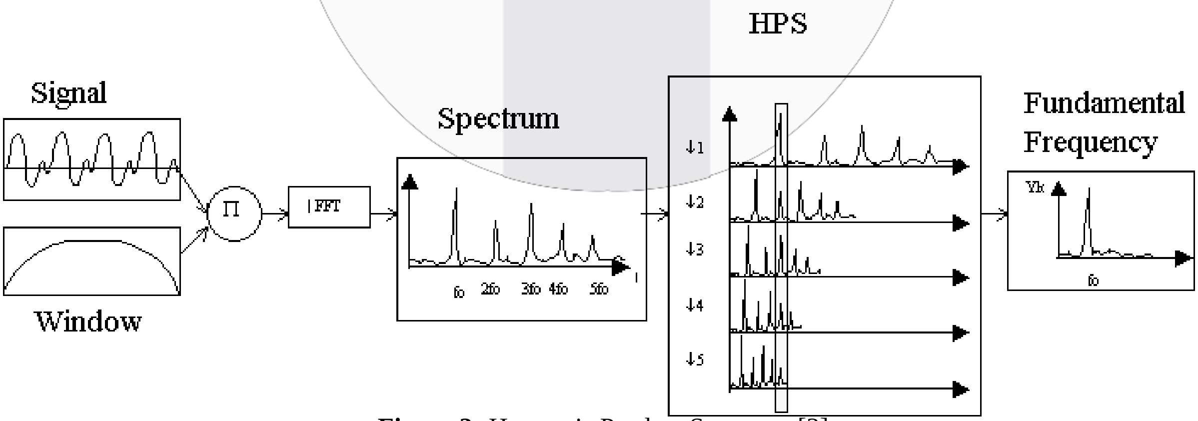

Figure 2: Harmonic Product Spectrum [3]

C. Fast Fourier Transform

A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. Fourier analysis converts time (or space) to frequency and vice versa; an FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. The FFT is a faster version of the Discrete Fourier Transform (DFT). The FFT utilizes some clever algorithms to do the same thing as the DTF, but in much less time [4].

D. k-Nearest Neighbor (KNN)

In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. In our research we use k-NN classification, therefore regression is not mentioned here.

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, it can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A shortcoming of the k-NN algorithm is that it is sensitive to the local structure of the dat**a [**5**].**

III. SYTEM OVERVIEW

The aim of this work is being able to identify random notes which are produced with saz. The system includes series of processes. Since the primary goal of this research is offline note classification, first thing to do is to record notes. All notes are recorded in Digital Signal Processing Labrotary of Telkom University in two different conditions: almost noiseless and noisy. The noisy condition is set by playing music as background noise while recording notes. Those notes are named as “Noisy Notes”. The Power level of the noise differs from note to note. The software used for rocording is “Cool Edit Pro”. The parameters used for recording are, 44100 Sample Rate, Stereo, 16 Bit Resolution. Once the notes ,so called raw data, are recorded we are ready to look for methods suitable for classification of those notes. There are several steps to classify the notes. Mainly: normalization, preprocessing, feature extraction, classification (see Fig, 3). The details of each of these steps are explained in following sections.

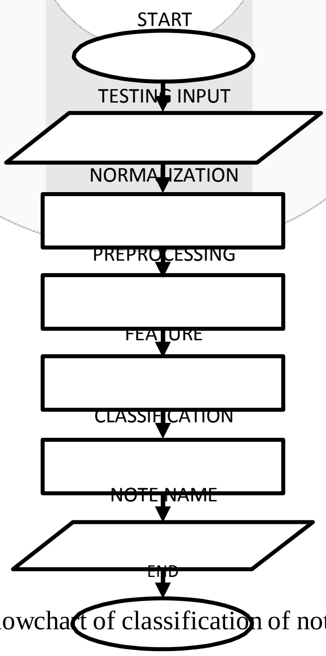

Figure 3: Flowchart of classification of notes of saz

A. Normalization

The raw audio file enters the system as wav file format with 44100 frequency rate and is normalized to a suitable data form for next steps. Normalization process includes picking channel, windowing and framing. Depending on audio format, audio may consist of one or many channels. Since our notes are recorded in stereo format, notes have two channels. Processing data with two channels means two times more computaion cost. Therefore the number of channels are reduced to one manually to save some computation time. The channel 1 and channel 2 usually includes different parts of audio. Whereas channel 1 includes the part with feature, channel 2 may include just noisy data or vice versa, which means we must pick the channel including the feature. Once the right channel is picked very next normalization step is framing and windowing. The frame length is 10000 and the window is between 10000th and 20000th sample. The reason to set the window between 10000th and 20000th is trying to pick the most stabilized area.

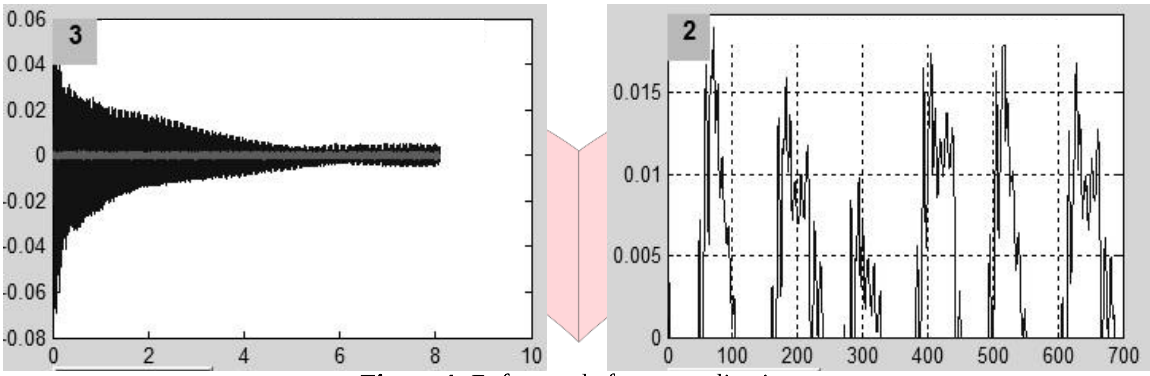

Figure 4: Before and after normalization

B. Preprocessing

Preprocessing phase is consist of two steps; Filtering and Fast Fourier Transform. The windowed data is filtered with band pass filter. Notes generated by Saz take place between 20 and 700 Hz. Therefore, a bandpass filter with first cut-off frequency 20 and second cut-off frequency 700 Hz is used. For some certain reasons it is impossible to filter all noises. One of the reasons is filter is not ideal. Another reason is the bandwidth of the filter also covers noises in same bandwidth but test results show that filter's performance is satisfying. In the feature extraction phase spectrums will be multiplied. Therefore the audio must be converted from time domain to frequency domain. The Fast Fourier Transform method is used to convert the audio from time domain to frequency domain.

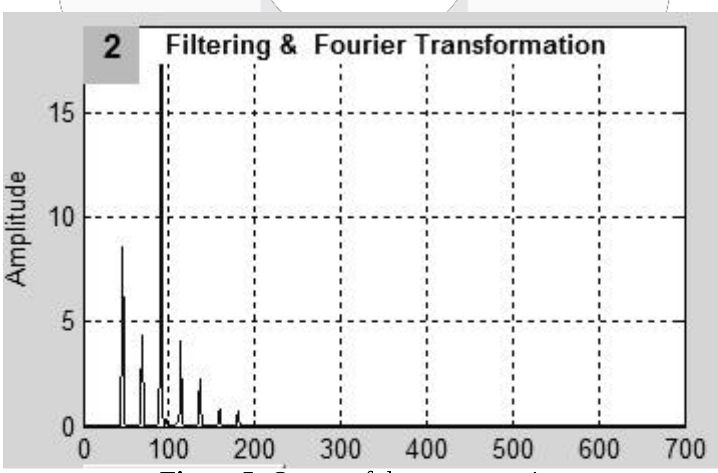

Figure 5: Output of the preprocessing

C. Feature Extraction

The features are extracted with Harmonic Product Spectrum Method. In Harmonic Product Spectrum Method data is down sampled several times and all down sampled data is multiplied together. As the result of that multiplication, all spectrums are reduced to 0 but the fundamental frequency.

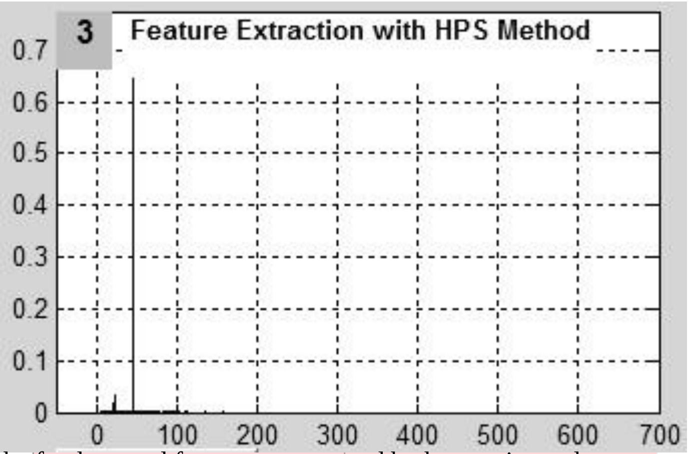

Figure 6: the fundamental frequency extarcted by harmonic product spectrum method

D. Classification

Classification is done by using k-NN method. Fundamental frequency is extracted with HPS. The distance between extracted feature and reference feature is calculated. The class of the training data is set according to nearest neighbor. This method works quite successfully.

IV. EXPERIMENTS AND DISCUSSION

8 different notes are used for experiments. The performance tests are held both in noisy and noiseless condition. 17 samples per each note are recorded for noiseless condition and 13 samples per each note are recorded for noisy condition.

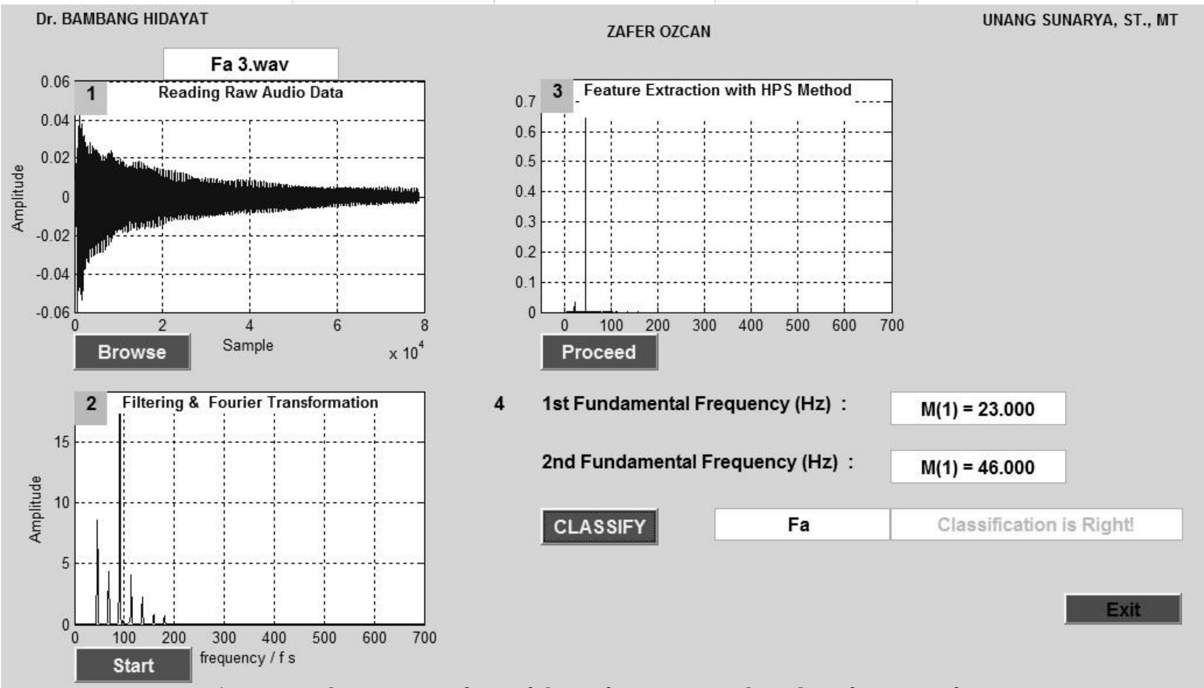

Figure 7: The user interface of the software created to classify notes of saz

Fundamental frequency of each note is found out with extracting the fundamental frequency of training data which are recorded in signal laboratory. The output of those noiseless signals are set as reference value for tests. The result of tests are saved.

Table 1: Test results for each notes

	0	1	2	3	4	5	6	7
DO	23-45	23-45	23-45	23-45	23-45	23-45	23-45	23-45
RE	19-37	19-37	19-37	19-37	19-37	19-37	19-37	19-37
MI	22-42	22-42	22-42	22-42	22-42	22-42	22-42	22-42
FA	24-46	23-46	23-46	23-46	23-46	23-46	23-46	23-46
SOL	37-49	25-49	25-49	25-49	25-49	25-49	25-49	25-49
I-SOL	25-50	25-50	25-50	25-50	25-50	25-50	25-50	25-50
LA	28-55	28-55	28-55	28-55	28-55	28-55	28-55	28-55
SI	30-60	30-60	30-60	30-60	30-60	30-60	30-60	30-60

In experiments we achieved 97.05 % accuracy for noiseless notes and 89.36 % accuracy for noisy notes. The software is able to extract the feature around 0.01 second. Overall computation time is around 0.0603 second.

Table 2: Test results of both noiseless and noisy condition

	F	R	T	%
NOISELESS	**4**	**132**	**136**	**97**
NOISY	**10**	**84**	**94**	**89**

Even though the percentage of the inaccurate detection is very low I would like to express some possible reasons of false detection. In our research we worked on 5 harmonics, which means the input signal is downsampled and multiplied 5 times. It is so hard to tell that we get the true fundamental frequency with 5 times downsampling but not 6 or 4. If the number 5 is somehow more than enough or not enough for input signal then the fundamental frequency becomes transparent to us. The second reason can be the lack of the musical skill of the person recording notes.

V. CONCLUSION AND FUTURE WORK

According to test results with noiseless notes we set the reference fundamental frequencies and started performance analysis. The experiments show that this algorithm based on harmonic product spectrum method successfully recognizes musical notes at the signal level. It assigns correct note names even in noisy condition with 89% accuracy and 97% accuracy in noiseless condition. Considering the simplicity and computation speed of the program, which is approximately 0.06 second, this is remarkable performance.

These results are very satisfying for us. Because the common accuracy level is around 95 % in classification for musical researches.

Still there are things might be done in the future to improve the capability of the research. One of the best thing to do is extending the capacity of the software with online note detection. Another future work is enabling the software to recognize all notes. Also the software can be improved to detect the notes in very noisy musical atmosphere. And the very important thing might be done is improving the software with automatic onset detection.

REFERENCE

[1] http://www.allaboutturkey.com/ [online] (accesed on Sept, 15th 2014)

[2] Arturo Camacho, A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music, University of Florida, 2007

[3] Philip McLeod, Fast, Accurate Pitch Detection Tools for Music Analysis, University of Otago, 30 May 2008

[4] University of Rhode Island Department of Electrical and Computer Engineering ELE 436: Communication Systems, 2006

[5] Aeberhard, S., Coomans, D., de Vel, O. Comparison of Classifiers in High Dimensional Settings. Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992.