Proceedings of 20th International Congress on Acoustics, ICA 2010

23-27 August 2010, Sydney, Australia

Feature for Musical Pitch Estimation from Simplified Auditory Model

Pat Taweewat

School of Electrical and Information Engineering, The University of Sydney, NSW, Australia

PACS: 43.75.Xz, 43.60.Lq

ABSTRACT

A simplified auditory model has been used for calculating an enhanced summary auto-correlation or ESACF, which can be used as a tool for musical pitch estimation from audio signal. The model itself is not only computationally efficient but its ESACF also shows a good result for single pitch estimation. However, using this ESACF for multiple pitch estimation seems to be very difficult to analyse because musical instruments usually have timbre variations even for the same kinds of musical instruments. By modifying this model, we can generate input features to use with neural network for assisting the process of multiple pitch estimation. Thus, each output of the neural network is mapped to each musical pitch and used to indicate each existing pitch probability. In our experiments, we generated data sets from recording of real musical instruments and used these data sets to train neural network and evaluate its performance. We compare performances of neural network between using of these proposed features and spectral features generated from audio spectrum. From the results, we found that the performances from these proposed features can be comparable with the features generated from audio spectrum and some experiments illustrated that these features yield better performances for musical instrument signals with slightly changes in their timbres.

INTRODUCTION

Musical pitch estimation is a task to find a correct pitch associated with audio signal at particular time frame. This task becomes complex when audio signal contains two or more pitch at the same time frame. We call this task specifically multiple pitch estimation. We found from the past work [1] that it is possible to use an artificial neural network for multiple pitch estimation and that work reports a good estimation results for synthesis signals. For detail of that work, the estimation system uses features from audio spectrogram or spectral features. Therefore, both time and frequency data are used for pitch estimation and make the system possible to detect musical note onset. However, frequency data may be sufficient if our task scopes only on multiple pitch estimation. Thus, in our experiments, we used only frequency data from audio spectrum.

Recently, a simplified auditory model [2] has been proposed for pitch estimation and found to be very useful for single-pitch estimation. In our work, we have an idea that integration between the simplified auditory model and neural network could improve multiple pitch estimation system. Thus, to prove our idea, we scope on performance comparisons between using features from audio spectrum and the simplified auditory model. Another aspect of our work is to find performances of our multiple pitch estimation system when performs estimation on real musical instrument signals. We also include more realistic situation when musical instrument signals have some variations. However, our investigation scopes on multiple pitch signals form single musical instrument only. The instruments included in our experiments are clarinet, oboe, horn, flute and trumpet.

ESTIMATION SYSTEM

In general, if we need to find musical pitch from audio signal, we should be able to define a mathematic function that maps between audio signal and its corresponding pitch as showed in Figure 1. Therefore, the audio signal is input variable and the pitch is output variable. For example, if we have single pitch signal, we can use auto-correlation as a mapping function. However, when relationship between input and output variables is very complex, such as multiple pitch signals, the way to define the correct mapping function is not straightforward.

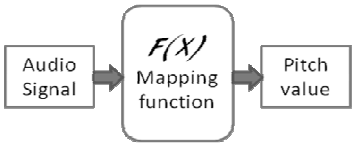

Figure 1. General idea for pitch estimation.

Fortunately, this problem can be solved using some kinds of an artificial neural network. A feed forward neural network has been proved to be universal function approximator [3] that can map between inputs and outputs by learning from pairs of input and output instances. The input instances are groups of attributes or what we call features (in this paper) whereas the output instances are groups of labels.

PROPOSE SYSTEM

INPUT UNIT

In our system, input unit is an interface between audio signals and input of neural network. In this unit, time domain audio signal is divided into a frame of 2048 samples and transformed into features for neural network by using three feature extraction methods: audio spectrum, equal temperament scale and simplified auditory model.

NEURAL NETWORK

There are many choices for types of the neural network. The past work [1] suggested using a time delay neural network. However, for our work, we consider a simple system since our goal is to investigate about features. Thus, we use a feed forward neural network as a mapping function between input features and output pitch labels. Each output pitch label is used to indicate each existing pitch probability. For convenience, we use 12-tet scale to reduce number of possible pitch frequencies. Thus, to cover all common musical pitch, the neural network has 88 outputs corresponded to 88 musical pitch frequencies. Each output shows each existing pitch probability.

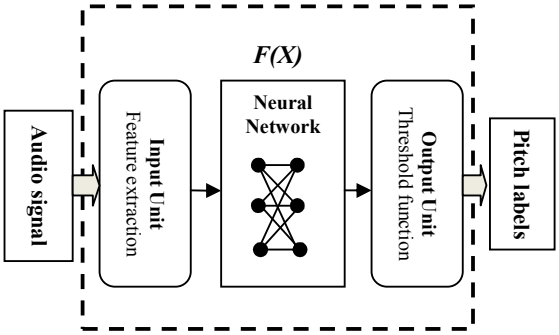

Figure 2. Proposed system.

Finally, the neural network contains three layers: input layer, hidden layer, and output layer. The specifications of the neural network are in table 1.

Table 1. Neural network specifications

Parameters	Values
Number of layers	3 [input, hidden, output]
Number of inputs	Depended on features
Number of hidden neuron	88
Number of outputs	88
Activation function	Hyperbolic tangent-sigmoid
Range of input values	[-1,1]
Range of output values	[-1,1]

OUTPUT UNIT

Neural networks with sigmoid activation function usually produce ambiguous outputs when input instances are associated with two or more possible cases. This is common because this activation function gives a probability rather than a hard decision. We prevent this situation by using the threshold function to select and assign final values to neural network outputs. In our system, the simple threshold function with threshold level=0.2 is used for each output.

FEATURES

Audio spectrum

Spectral features are features generated using the concept of audio spectrum. We know that musical pitch and frequency components have relationship in someway. Thus, we use the neural network as an engine to map audio spectrum to musical pitch. In our work, we limit one frame of audio signal to 2048 samples. Therefore, to generate spectral features, we take FFT on these 2048 samples and keep only FFT magnitudes. Since FFT magnitudes are symmetrical, we keep only the first 1024 values as spectral features.

Equal temperament scale

In western music, it is common to define musical pitch as frequency on 12 tone equal temperament scale or 12-tet. We know that frequency spaces on this scale are non-linear and exponentially distributed. Therefore, we can reduce number of spectral features by scaling linear frequency spaces of FFT using 12-tet scale. By this way, only 88 features are used since 88 pitch frequencies cover frequency ranges of common musical instruments. Equation (1) shows relationship between pitch in term of note number and frequency.

$$f = 27.5 \times 2^{\left(\frac{n-1}{12}\right)} \quad (1)$$

Where f = frequency in Hz and n = note number.

However, practically, we found that frequency resolutions of FFT with 2048 points are too low for frequency lower than 387 Hz. Thus, we directly use FFT frequency spaces for the first 19 frequency bins whereas the rest of frequency bins are scaled to 12-tet scale. By this way, we have only 61 features as showed in Figure 3.

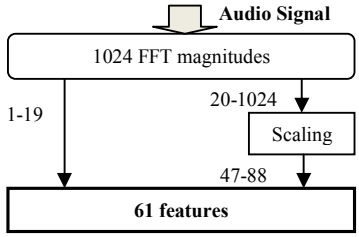

Figure 3. Features from equal temperament scale.

Simplified auditory model

In the recent works [4-5], we found that human auditory system has a process that can be modelled as non-linear multi-channel filter bank. The term "non-linear" means that centre frequencies are non-linearly distributed and signal levels of all filter channels are compressed. To estimate musical pitch from this model, we need to calculate auto-correlation of signal from each filter channel and then all auto-correlations from all filter channels are summarised to final decision.

The problem with this model is that we need to process many channels of filter bank. This leads to computational expensive. Fortunately, research work [2] shows that we can use filter bank with only two channels to find the pitch. The two channels are low and high frequency channels. This kind of filter bank is used to form simplified auditory model. However, this model has slightly different from full auditory model in that only signal from high frequency channel is non-linearly processed. To make the model simple, non-linear operation is replace by half wave rectifier.

In the original work [2], the outputs of both low and high frequency channels are combined for calculating ESACF and used for pitch estimation. In the same way as auto-correlation function, position of peak from ESACF shows frequency period of considering signal. Thus, we can estimate the pitch from this peak.

However, we found difficulty to use ESACF for multiple pitch estimation because of two reasons. First, ESACF calculation is based on time-domain resolution that restricted to sampling frequency. Therefore, two peaks cannot be separated well at high frequencies. Second, when signal has two or more musical pitch ESACF usually generates many unrelated peaks that are hard to be interpreted.

For those reasons, in our work, we do not calculate ESACF and use data from frequency domain instead. To generate features from simplified auditory filter bank, we take FFT on each output of filter bank. We keep only FFT magnitudes then scale the frequency spaces using 12-tet scale to reduce number of features. Thus, by this way, we have 122 features. The process to generate these features is showed in Figure 4.

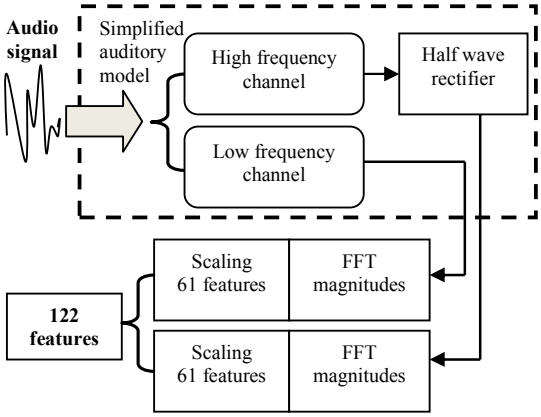

Figure 4. Features from simplified auditory model

The filters in the model are implanted using 4th order gamma tone filters. Cut-off frequencies are illustrated in table 2.

Table 2. Cut-off frequencies in our model.

Cut-off frequencies	Values
High frequency channel	[1 KHz,10 KHz]
Low frequency channel	[15 Hz,1 KHz]

Haft-wave rectifier can be approximately calculated using equation (2).

$$HWR = \frac{X + |X|}{2} \quad (2)$$

Where *HWR*=signal from half waver rectifier and *X*=input signal to half wave rectifier.

Magnitude scales

Magnitude scales have an important effect on the estimation performances. Our auditory system is able to adapt to wide dynamic levels of various audio sources. We approximate this level adaptability using logarithm scale. In our experiments, we need to know how well this approximation can improve our estimation system. Thus, for comparison, we generated two sub types of features using linear-magnitude and log-magnitude scales during evaluation phase.

TRAINING METHOD

Musical note samples

Musical note samples are time domain audio signals of each musical note. In our experiments, these samples were prepared using real musical instrument recordings from RWC database [6]. We define starting position of each musical note by using position where absolute magnitude of each musical note signal reaches 60% of its highest magnitude. By this way, each sample contains only or at least almost stable (sustain) part of each musical note signal.

Training sets

During training phase, we generated five separated training sets for five instruments: Clarinet, Oboe, Horn, Flute, and Trumpet. Each of these training sets contained 500 audio mixtures of one, two, three and four musical notes. Thus, total mixtures per training set were 2000. We generated each mixture by randomly selecting from prepared musical note samples as mentioned above. After that, we mixed these selected musical samples at 1:1 ratio.

Cross validation

To evaluate the performance of neural network, we using k-fold cross validation to partition each of training sets into two sub sets: actual training and validation sets. In our experiments, we chose k=3 to prevent long training time. Although k=10 is common in many literature, we found that k=3 could be better to see generalization of our system than k=10. This is because k=3 produces less number of instances for training (67%) and more number of instances for validation (33%).

During training phase, we used scale conjugate gradient algorithm to train the neural network since this algorithm is usually suitable for complex problems. However, this algorithm is sensitive to initial random weights. Thus, to prevent this problem, in our experiments, we trained each neural network for three times and chose the neural network that produced the best performance for each fold.

EVALUATIONS

Performance calculations

To evaluate performance of features, we used three common calculations: precision, sensitivity (recall), and F1-measure. The following equations show these calculations.

$$prec = \frac{Tp}{Tp + Fp} \quad (3)$$

$$sens = \frac{Tp}{Tp + Fn} \quad (4)$$

$$F1 - measure = 2 \times \frac{prec \times sens}{prec + sens} \quad (5)$$

Where *Tp* = number of true positive, *Fp* = number of false positive, and *Fn* = number of False negative.

We can see that F1-measure represents both precision and sensitivity. Therefore, to save paper's space, we only show results from F1-measure. All results were averaged from 3 neural networks since we trained them using 3-fold cross validation.

Test sets

We also generated test sets using real recordings from RWC database [6]. All samples in the test sets were completely unseen by the neural network because we generated the training and test sets separately.

In the same way as training sets, we generated the test sets separately for five instrument signals. Each test set has 2000 samples from 500 of one, two, three and four note mixtures. However, to make our results more realistic, we generated the test sets by using different samples from the training sets. The ways to prepare these samples were depended on types of test sets. In our experiments, we had two types of the test sets: the test sets generated from samples with varying the starting position and samples with different instrument manufacturers. Thus, we have 10 test sets for five instrument signals. We call these two types of the test sets: situation I and situation II. Both two situations are explained as follow.

Situation I: varying the starting position

In real situation, amplitudes and phases of audio signals before mixing can be arbitrary. A good multiple pitch estimation system should be possible to produce good estimation results even when this situation occurs.

To evaluate our system using this situation, we prepared musical note samples by finding the starting position of each note signal in the same way as we did for the training sets. However, we randomly vary the position away from the starting position (not more than 185 msec). All samples in these test sets were not normalized after varying the starting position. Thus, samples in these test sets were not mixed at 1:1 ratio but arbitrary ratio depended on randomly selected position of note signal.

Situation II: different instrument manufacturers

In this situation, we generated musical note samples from the same types of musical instruments but changed the manufacturers. In the same way, if multiple pitch estimation system has good generalization, it should perform well in this situation. Therefore, we evaluated our system in this situation by preparing the samples in the same way as in situation I. However, the samples in these test sets were from audio samples recorded with different instruments manufacturers from situation I.

RESULTS AND DISCUSSIONS

For result comparisons, we show F1-measures using bar charts with abbreviations for convenience. The abbreviations used in these charts are summarised in table 3 and table 4.

Table 3. Feature types and their abbreviations.

Abbreviations	Feature types
A	Audio spectrum
B	Equal temperament scale
C	Simplified auditory model
D	Audio spectrum (log magnitude)
E	Equal temperament scale (log magnitude)
F	Simplified auditory model (log magnitude)

Table 4. Musical instrument types and their abbreviations.

Abbreviations	Musical instrument types
Cl	Clarinet
Ob	Oboe
Ho	Horn
Fl	Flute
Tr	Trumpet

Training performances

We need to know how well the neural network learns from the training sets and our training method. Thus, we evaluated our system by using the training sets as test sets. The results, in Figure 5, show that all features produce high estimation scores. F1-measures from all instrument signals are higher than 0.94.

Situation I

From the results, in Figure 6, we found that our system performs quite well even amplitudes and phases of signals are slightly changed. Most of all features produce F1-measures higher than 0.65 and interestingly, F1-measures are higher than 0.8 (except for French horn) when the neural network used features from the simplified auditory model. This is for both linear and log magnitude scales.

Situation II

This is the hardest situation for our system because all degrees of signal variations are very high. In Figure 7, we found that the neural network performs poorly for features generated from audio spectrum with linear magnitude scale whereas features with log-magnitude scale seem to improve the performances slightly. The important point is that, features generated from simplified auditory model still boost the neural network performances in this situation especially when using log-magnitude scale. Almost all F1-measures from the system using this type of features are higher than 0.7 except when we evaluated the system with signals from French horn. The reason should be frequency range of French horn since in all musical instruments in the test, French horn produces the lowest note.

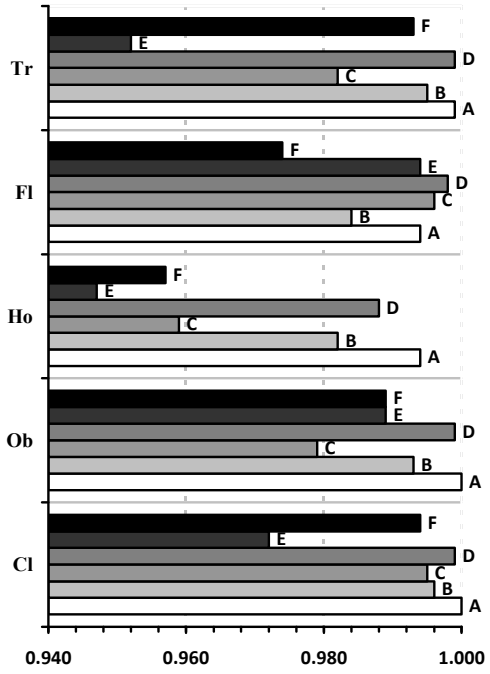

Figure 5. Evaluations using the training sets.

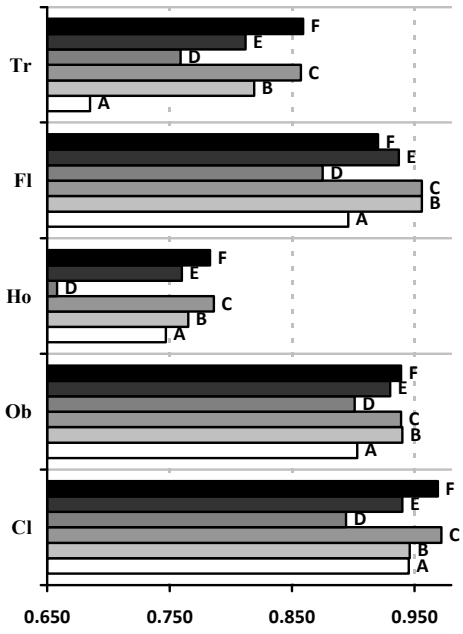

Figure 6. Evaluations using situation I.

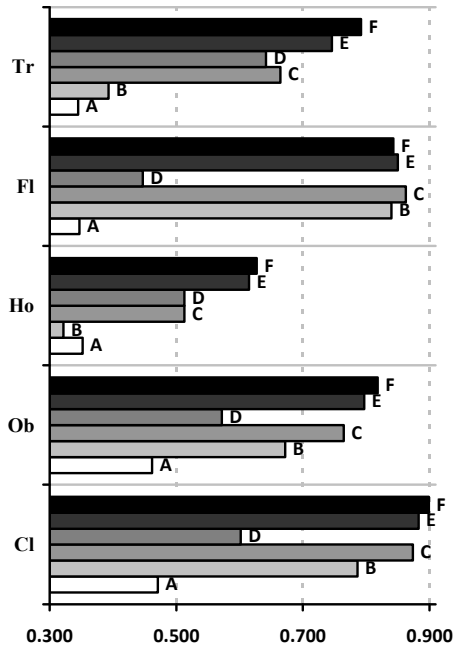

Figure 7. Evaluations using situation II.

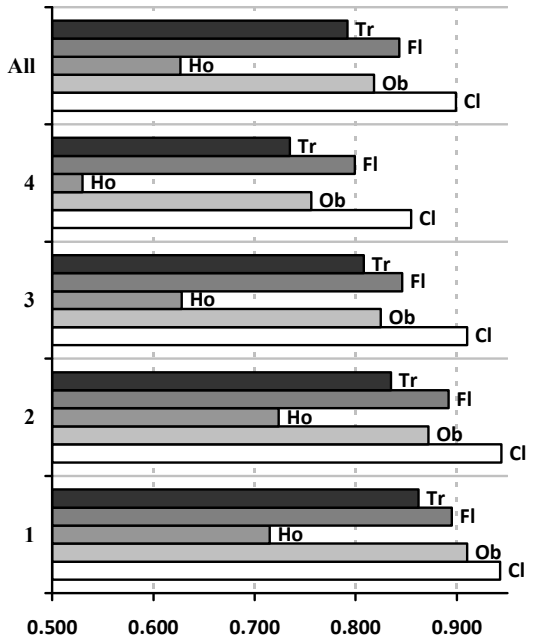

Figure 8. Results using features from simplified auditory model for particular polyphony numbers.

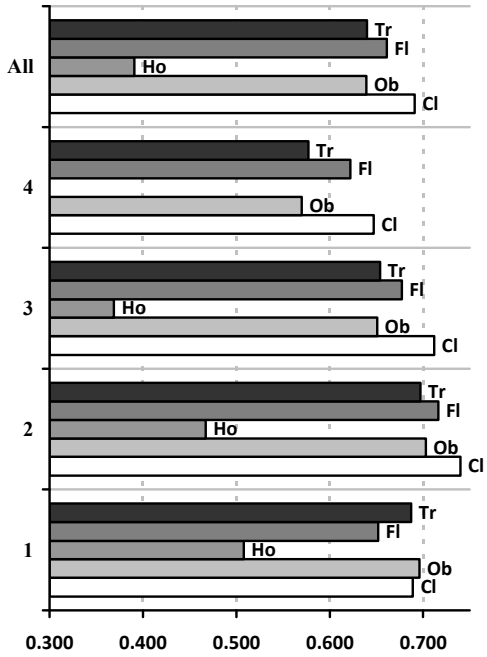

Figure 9. Results using features from simplified auditory model when signals contain -10 dB reverberation.

PERFORMANCE SUMMARY

To summarise results from all evaluation cases, we use number of outperforming scores for each type of features. The results are illustrated in Table 5 using the same abbreviations as explained above. It is clearly that when using the neural network with features from simplified auditory model (log-magnitude), the system performs the best for situation II.

Although these features cannot make the system perform well for situation I and the training sets, overall results still show that features from simplified auditory model (log-magnitude) are the best.

Table 5. Performance Summary.

Sets	Feature Types					
	A	**B**	**C**	**D**	**E**	**F**
Training sets	**4**	0	0	2	0	0
Test sets Situation I	0	2	**3**	0	0	1
Test sets Situation II	0	0	1	0	0	**4**
Total	4	2	4	2	0	**5**

ADDITIONAL EXPERIMENTS

Situation II + reverberation

We used SIR software [7] to add reverberation to the situation II test sets and evaluated our system with features from simplified auditory model (log-magnitude). The impulse response was "cAPS-ccp-xy3" from [8]. For detail comparisons, we show the results for each particular polyphony number (number of musical note per mixture sample). The results without reverberation are in Figure 8 whereas Figure 9 shows results from the same test sets with -10 dB reverberation added. As we can see, when the numbers of notes are less than four, the results from signals with reverberation are still higher than 0.6 for almost all instrument signals (except for French horn).

CONCLUSION

In this paper, we proposed musical pitch estimation system using simplified auditory model and feed forward neural network. The system can be use to estimate multiple pitch from audio signal generated from single type musical instrument. Our extensive scope of this paper is to investigate performances of the neural network when different types of features are used. From experiments, we found that using features from simplified auditory model produce better musical pitch estimation when timbres are slightly varied.

REFERENCES

[1] A. Pertusa and J. M. Inesta, "Polyphonic music transcription through dynamic networks and spectral pattern identification," presented at the Internetional Conference on Artificial Neural Networks in Pattern Recognition Acoustics, Florence, Itary, 2003.

[2] T. Tolonen and M.Karjalainen, "A Computationally Efficient Multipitch Analysis Model," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 8, p. 9, 2000.

[3] S. V. Kartalopoulos, *Understanding Neural Networks and Fuzzy Logic*: IEEE Press, 1996.

[4] T. Irino and R. D. Patternson, "A Dynamic Compressive Gammachirp Auditory Filterbank," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 14, p. 11, 2006.

[5] A. Klapuri, "Multipitch Analysis of Polyphonic Music and Speech Signals Using an Auditory Model," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 16, p. 12, 2008.

[6] M. Goto, *et al.*, "RWC music database," in *ISMIR*, 2003.

[7] C. Knufinke. (2005, *SIR Impulse Response Processor*. Available: www.knufinke.de/sir

[8] J. Johnson. (2005, *Impulse response of 1400 seat church*. Available: www.noisevault.com