MUSIC TRANSCRIPTION USING AN INSTRUMENT MODEL

Jun Yin, Terence Sim, Ye Wang, Arun Shenoy
National University of Singapore, School of Computing, Singapore 117543
{yinjun, tsim, wangye, arunshen}@comp.nus.edu.sg

ABSTRACT

We introduce a method to transcribe music with the help of an instrument model. One of the most important and difficult problems in music transcription is polyphonic pitch estimation. Common pitch estimation algorithms reported in the literature tend to have errors in the following three situations: missing fundamental, missing harmonics, and shared frequencies. We believe an instrument model can make pitch estimation more robust in these three situations and thus can help to improve music transcription accuracy. We devise a spectrum subtraction algorithm to transcribe single and multiple instrument polyphonic music.

1. INTRODUCTION

Automatic music transcription deals the transformation of acoustic musical signals into a symbolic representation such as MIDI or a musical score which in principle, could then be used to recreate the musical piece [12]. This has many applications. Music in the notation is useful in many applications. It is more compact to store and more efficient to process and transmit than acoustic audio. Transcription can be used as a visualization of media players, which displays the music score during the playback. It can also be used to monitor students playing musical instruments, by transcribing the music played by the student and evaluating it against the standard score. Transcription also plays an indispensable role in melody based music retrieval, such as query by humming [5].

2. PREVIOUS WORK

Over the years, considerable work has been done in music transcription. Some of the earlier work [6][10] has been restricted to monophonic music. Marolt and Privosnik designed a system [11] to transcribe polyphonic piano music using a combination of partial track extraction, onset detection and note recognition. Another system to perform piano music transcription using a Hidden Markov Model approach has been presented by Raphael [3]. Martins and Ferreira [7] developed a general polyphonic music transcription using a combination of harmonic structure tracking and a post processing stage that attempts to identify the best trajectories to represent the true musical notes played. Goto [8][9] proposed an Expectation-Maximization algorithm based technique for estimating the fundamental frequency (F0) of melody and bass lines in real world musical audio signals and CD recordings. Tolonen and Karjalainen [13] have developed a multiple-F0 estimation of musical sounds in using a modified version of the unitary pitch model. Klapuri et al [1] proposed an iterative spectrum smoothing and subtraction algorithm for multiple pitch estimation of concurrent musical sounds, which is considered one of the best pitch estimation algorithms today.

3. OUR METHOD

The ability to transcribe polyphonic music usually depends on the pitch estimation module of the transcription system. However, we found that common polyphonic pitch estimation algorithms tend to have errors in the following three situations:

- missing fundamental
- missing harmonics
- shared frequencies

Some earlier algorithms assume the fundamental exists, and consider all the frequency peaks in the spectrum as pitch candidates. These will not find sounds whose fundamental is missing.

Some algorithms based on statistics consider a frequency as pitch only when there are enough harmonics to support the fundamental. Therefore they will not work in the case that some harmonics of the sound are missing.

Sharing frequencies is one of the most difficult problems that polyphonic algorithms face. Algorithms often have trouble determining how much of the shared frequency belongs to each note. The simplest example is A3 and A4, whose frequencies are 220*k* and 440*k* (*k*=1, 2, 3…) All the frequencies of A4 coincide with harmonics of A3. This ambiguity causes some algorithms to fail to detect A4, because all the frequencies of A4 can be considered as harmonics of A3. Klapuri et al [1] tried to solve this problem by applying the spectral smoothing principle.

We believe that an instrument model which characterizes the harmonic structure [7] of the instrument can help our pitch estimation algorithm to be more robust in these three situations.

0-7803-8874-7/05/$20.00 ©2005 IEEE

3.1. Instrument Model

3.1.1. Building the Model

The instrument model contains the harmonic structure information of the instruments used in the music. A sample from each instrument is required to build the model. An amplitude spectrum of each instrument sample is created using FFT. The amplitude spectrum is divided into semitone bands (say, 88 bands from A0 to C8), whose central frequency is the note frequency, and whose bandwidth is a semitone (half of a semitone higher and lower than the central frequency). The band energy spectrum is computed using the following formula:

$$Z[i] = \sum_{k=LB(i)}^{UB(i)} (Y[k])^2 \quad i = 1..88 \quad (\text{A0..C8})$$

where i is the band index, Z is the band energy ($Z[1]$ is the energy of A0, $Z[2]$ is the energy of A#0, etc), Y is the amplitude spectrum, $LB(i)$ and $UB(i)$ are the lower bound and upper bound of band i.

If the sample note is played at pitch i, $Z[i]$ is the energy of the fundamental, which may be low or even zero. According the frequency relationship of the harmonics to the fundamental, the first, second, third…16th harmonics will lie in $Z[i+12]$, $Z[i+19]$, $Z[i+24]$…$Z[i+48]$ respectively. Assuming the harmonics beyond the 16^{th} harmonic are weak and can be ignored, this 49 number vector $Z[i..i+48]=I[0..48]$ will carry the information of the harmonic structure, and characterize the feature of this instrument. We extract this feature vector for each instrument which is used in the music. The vector list is the instrument model.

3.1.2. Use of the Model

We assume the harmonic structure of the musical sound of the instrument is the same regardless of the pitch and transient of the sound. With an instrument feature vector I, the band energy spectrum of any note from that instrument can be easily generated. For a note with volume a and pitch p, the spectrum is simply obtained by magnifying the vector I by ratio a, and then shifting it to position p, as shown in the following formula:

$$Z[i] = \begin{cases} a \cdot I[i-p] & , \quad i \in [p..p+48] \\ 0 & , \quad otherwise \end{cases}$$

This formula can be regarded as a "spectrum generation function", with the form below:

$$Z = F(I, a, p)$$

3.2. Pitch Estimation Using Instrument Model

3.2.1. Single-instrument Pitch Estimation

Single-instrument pitch estimation with an instrument model refers to the pitch estimation of a frame in the single-instrument input music. More precisely, we define:

Input: Music band energy spectrum Z_M; Instrument feature vector I.

Output: Volume and pitch pairs (a_i, p_i), $1 \le i \le n$, n is the number of notes, so that $\left\| Z_M - \sum_{i=1}^{n} F(I, a_i, p_i) \right\|^2$ is minimized. Note that the number of notes n is also unknown.

This definition is based on the principle that if notes are correctly detected and their volumes and pitches are correctly estimated, their sum energy in each band should be equal or similar to the energy of the same band in the music.

We devise a spectrum subtraction algorithm to solve the above problem. We start from the lowest note A0, and match the instrument feature vector $I[0..48]$ to a section of the music spectrum $Z_M[1..49]$. The volume of note A0 is estimated by finding such a coefficient a_1 that $Z_M[1..49]$ approximately equals to $a_1 I[0..48]$ (minimum sum square error). If a_1 is greater than a threshold, the note $(a_1, 1)$ is created. Otherwise that note is considered absent in the music. If the note $(a_1, 1)$ is created, its spectrum is estimated as $F(I, a_1, 1) = a_1 I[0..48]$, which is subtracted from the music spectrum $Z_M[1..49]$. Then we slide the instrument feature spectrum $I[0..48]$ to match the remainder music spectrum $Z_M[2..50]$, to estimate the volume of note A#0. If it satisfies the conditions and the note is created, its spectrum is subtracted from the music spectrum as well. This "match, subtract, slide" process continues until the volumes of all the 88 notes are estimated.

We start to match from low pitch to high pitch because a higher pitch band may contain the energies from the harmonics of lower pitch notes. By matching the lower pitch first, these energies which belong to the lower pitch notes can be subtracted from the spectrum before higher pitch is matched.

In each match, we are trying to find a coefficient a_i to minimize the error between $Z_M[i..i+48]$ and $a_i I[0..48]$. This can be solved by standard linear regression [4]:

$$a_i = \frac{\sum_{k=0}^{48} (I[k] \cdot Z_M[i+k])}{\sum_{k=0}^{48} (I[k])^2}$$

In practice, we find that the fundamental and lower harmonics are more stable then higher harmonics, which is more important in the match. In order to estimate the volume a_i more accurately, we apply a weighted linear regression:

$$a_i = \frac{\sum_{k=0}^{48} (W[k] I_1[k] \cdot W[k] Z_M[i+k])}{\sum_{k=0}^{48} (W[k] I_1[k])^2}$$

where W is a weight vector. Higher weights are given to the fundamental and lower harmonics, for example,

$W[0]=0.6$, $W[12]=0.3$, $W[19]=0.1$, and $W[k]=0$ when $k\neq 0$, $k\neq 12$, $k\neq 19$.

Our algorithm is more robust in the three situations mentioned at the beginning of Section 3. In the case of missing fundamental, the fundamental can still be located correctly by matching the harmonics in the music spectrum with the harmonics in the instrument feature vector. The case of missing harmonics is the same. In the case of shared frequencies, after the lower pitch note is detected, the amount of frequency components that only belong to that note is removed from the spectrum. Therefore the higher pitch note can still be detected.

3.2.2. Multi-instrument Pitch Estimation

Multi-instrument pitch estimation with instrument model is defined as the following:

Input: Music band energy spectrum Z_M; Instrument feature vectors I_1, I_2, …, I_m, m is the number of instruments used in the music.

Output: Volume, pitch and instrument pairs (a_i, p_i, q_i), $1\leq i\leq n$, n is the number of notes, so that $\left\| Z_M - \sum_{i=1}^{n} F(I_{q_i}, a_i, p_i) \right\|^2$ is minimized.

Our spectrum subtraction algorithm for single-instrument can be easily extended to multi-instrument. Based on the fact that several instruments can play notes of the same pitch, in each match i, the section of music spectrum $Z_M[i..i+48]$ is matched with linear combination of all the instrument feature vectors in the model I_1, I_2, …, I_m, in order to find out the volumes $a_{i,1}$, $a_{i,2}$, …, $a_{i,m}$, so that the error between $Z_M[i..i+48]$ and $\sum_{j=1}^{m} a_{i,j} I_j[0..48]$ is minimized. The match can be solved by standard linear regression or weighted linear regression similarly.

3.3. System Implementation

We developed a complete, workable music transcription system based on our pitch estimation algorithm, which transcribes polyphonic music, outputs a note table and generates a MIDI file. The input music and instrument sample wave files are re-sampled at 22 kHz 16-bit. An 8192 point FFT is used to create amplitude spectra of the music and the instrument samples. Blackman window is applied to each frame, and the frame shift is 512 samples. Therefore the frequency resolution is 2.69 Hz and the time resolution is 23ms. Band energy spectra are computed amplitude spectra. Instrument feature vectors are extracted. For each frame in the music spectrograph, our spectrum subtraction algorithm is used to estimate the pitch. After that, a common onset detector is used to detect the onset of the notes. From each onset location, subsequent frames are tracked until the note volume fades below a threshold, to compute the duration of the note. Then a potential note is created in the note table. An example of the output note table is shown in Table 1.

Onset (frame index)	Duration (frame number)	Volume a	Pitch p	Instrument q
25	33	1.1084	40	1
77	32	0.6753	44	2
129	32	1.1368	44	1
129	29	0.6714	47	2

Table 1: A sample note table

The note table is then converted into an MIDI file, which can be played back directly. Our transcription module is written in Matlab and our conversion module is implemented in C++.

3.4. System Evaluation

We use the following method to evaluate our system. A music MIDI file which contains the ground truth notes is created manually. The music MIDI file is played back with Microsoft Wavetable Synthesizer and recorded into a wave file. Microsoft Wave Table is also used to generate instrument sample wave files. Then the music wave file along with the instrument wave files is given to our system, and an output MIDI file is obtained. Finally the notes in the output MIDI file are compared with those in the input MIDI file.

The pitch range of notes in our manually created MIDI files is from A2 to C8, due to our FFT frequency resolution. The number of simultaneous notes (polyphony) is from 1 to 10. The instruments used are piano and clarinet, which have a stable harmonic structure.

In [2], Klapuri has compared his test results in pitch estimation from [1] against other various systems such as [8] and [13] and shown it to be better. Thus in this work, we evaluate our system using Klapuri's results as a baseline. We also use NER (note error rate) to measure the pitch estimation accuracy of our system. NER is defined as the number of errors in divided by the number of ground truth notes. A note is considered correctly detected of the pitch is correct, regardless of the volume. Each of the three situations is considered as one error: detecting an extra note (insertion error), detecting one fewer note (deletion error) and detecting a note of different pitch (substitution error).

Through our tests, we find that the NER of our system with polyphony from 1 to 10 is 0%, 0%, 0%, 1.1%, 1.7%, 3.8%, 4.9%, 6.1%, 8.3%, 11%, respectively, which is lower than those of Klapuri's system. Their NER with polyphony from 1 to 6 is 2.1%, 2.4%, 3.8%, 8.1%, 12%, 18%, respectively. This is shown in Figure 1.

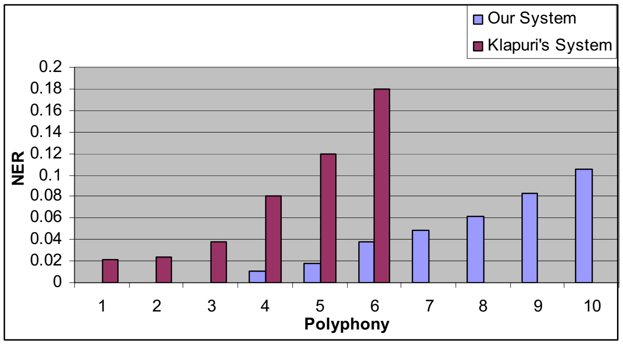

Figure 1: Accuracy comparison between our system and Klapuri's System

This result shows that, when the sounds in the music have stable harmonic structure, the accuracy of music transcription can be improved by using the instrument model, which serves as prior knowledge. It is to be noted that our test set consists of only two instruments of stable harmonic structure as compared to Klapuri's test set which consists of 26 different musical instruments and sung vowels. Therefore, we conclude from our test results that our system though more restrictive, shows a much higher performance for instruments having a stable harmonic structure.

Further, the time complexity of our system is very low, which is suitable for real-time applications. If we divide the amplitude spectrum to *m* bands, there are *m* matches in the pitch estimation of a single frame. If the music is *n* frame long, there are *mn* matches in total. Since *m* is a constant, the time complexity is O(*n*), which is linear to the length of the music.

4. CONCLUSION AND FUTURE WORK

In this paper we proposed a new method to transcribe music using an instrument model. The instrument model makes our polyphonic pitch estimation more accurate and robust. Our test result shows that this new method outperforms the other methods in the case of stable harmonic structure.

Our current pitch estimation algorithm extracts only one feature vector from each instrument, and uses it to match the band energy spectrum in the music. This is based on the assumption that the harmonic structure of an instrument is the same regardless of the pitch and transient. However, we find that the harmonic structure does change slightly with pitch and time. In the future, we will try to build a pitch and time varying instrument model, in order to achieve even higher music transcription accuracy.

Moreover, our current system requires the user to input the instrument samples, and then use these samples to build the instrument model. We will research on how to analyze the harmonic structures of the instruments directly from the music, so that the instrument model can be built from the music itself.

5. REFERENCES

[1] Anssi Klapuri, Tuomas Virtanen, Jan-Markus Holm. "Robust Multipitch Estimation for the Analysis and Manipulation of Polyphonic Musical Signals". *COST-G6 Conference on Digital Audio Effects*, December 7-9, 2000.

[2] Anssi P. Klapuri. "Automatic Transcription of Music". *Proceedings of the Stockholm Music Acoustics Conference*, August 6-9, 2003, Stockholm, Sweden.

[3] Christopher Raphael. "Automatic Transcription of Piano Music". In *Proc. ISMIR 2002*, pages 15–19.

[4] George A. F. Seber, Alan J. Lee. Linear Regression Analysis, 2nd Edition. ISBN: 0471415405

[5] Goffredo Haus and Emanuele Pollastri. "An Audio Front End for Query-by-Humming Systems". In *Proceedings of International Symposium on Music Information Retrieval*, 2001.

[6] James A. Moorer. "On the segmentation and analysis of continuous musical sound by digital computer". *PhD thesis*, Department of Music, Stanford University, Stanford, CA, May 1975.

[7] Luis Gustavo Martins and Anibal Ferreira. "PCM to MIDI Transposition". *112th Audio Engineering Society*. Munich, Germany, May 10-13, 2002.

[8] Masataka Goto. "A Predominant-F0 Estimation Method for Real World Musical Audio Signals: MAP Estimation for Incorporating Prior Knowledge about F0s and Tone Models". In *Proc. Workshop on Consistent and reliable acoustic cues for sound analysis*, Aalborg, Denmark, September, 2001.

[9] Masataka Goto. "A Predominant-F0 Estimation Method for CD Recordings: MAP Estimation Using EM Algorithm for Adaptive Tone Models," Proc. ICASSP 2001.

[10] M. Piszczalski and B. A. Galler. "Automatic Music Transcription", *Computer Music Journal 1(4):24-31*, November 1977.

[11] Matija Marolt and Marko Privosnik. "SONIC: A System for Transcription of Piano Music". In *Advances in automation, multimedia and video systems*, WSES Press, 2001.

[12] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler. "Automatic Music Transcription and Audio Source Separation". *Cybernetics and Systems, 33(6):603–627*, 2002.

[13] Tolonen, T. and Karjalainen, M. "A computationally efficient multipitch analysis model". *IEEE Trans. Speech Audio Processing, Vol. 8, No. 6, pp. 708-716*, November, 2000.