Transcription of Piano Music

Rudolf BRISUDA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xbrisuda@is.stuba.sk

Abstract. Music transcription can be solved in several ways. We present the-state-of-the-art in automatic polyphonic transcription and solution of automatic pages turning for piano music. We analyze problems of music transcription which could be used for this purpose. We focus on keystroke detection (Note Onset Detection based on Spectral flux) and detection of tones (simple and computationally efficient method to polyphonic pitch detection based on Summing Harmonic Amplitudes) in this keystroke. Whereas detection of keystroke often fails to track position in song, we propose an algorithm which corrects position within the song by polyphonic pitch detection. Proposed algorithm repairs Spectral flux with Polyphonic pitch detection algorithm and it outperforms the Spectral flux itself.

1 Introduction

Pianists have often problem with turning pages while playing songs. Therefore, they often missed a part of the song because they use the hand to turn the page. They have to learn the songs by heart if they want to play flawlessly or not use "ninja moves" to turn the pages. Many musicians use to store and display music sheets by the tablets which provide new possibilities. For example, algorithm of music transcription should be able to determine where the pianist in the song is and the algorithm could automatically assess when to turn the page.

There are hardware solutions based on foot pedals. One problem still remains, musician still need to pay an attention to additional device. We focus on automatic turning of the pages by using a microphone. One of the advantages of the use of this algorithm could be a higher portability and no need of additional devices.

Our aim is to develop a solution which will analyze the sound captured by a microphone in real-time. We focus on certain types of algorithms belonging to music transcription and we try to solve this problem in the simplest way. For this purpose, we decide to use the algorithm to onset (mainly) and Polyphonic Pitch Detection (PPD). First, piano keystrokes will be detected with some accuracy and it will be corrected with detected notes. Keystrokes will be tracked in the played song by comparing sound data with input data of music sheets and when they reach end of the page, the page will be turned. Both of the algorithms operate with some accuracy and tracking song only by one approach provides poor results.

---

* Bachelor study programme in field: Informatics
Supervisor: Andrej Fogelton, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 331–336.

2 State of the art

Pitch detection algorithms are designed to detect pitch or fundamental frequencies from sound signals (e.g. music or speech). These algorithms have been developed primarily with the interest in speech recognition. There are many complex methods which reflect this nontrivial problem [4,8,10,11,14]. The algorithms can be divided into the following categories: time domain method, frequency domain method, combination of time and frequency methods and models of human ears.

Time domain methods (TDM) operate directly with the input signal as a fluctuating amplitude. They look on the waveform with the aim to find repeating patterns which indicate periodicity. The principle of the frequency domain method involves dividing of the input signal into the frequencies. These frequencies represent the spectrum which shows their strength. The typical analysis include Short Time Fourier Transform (STFT) [13]: division of signal into segments, applying window and subsequently on each segment performing Fourier Transform. This shows peaks which may correspond to pitches (fundamentals frequencies), harmonics (integer multiples of the fundamental frequencies or redundant parts. The aim is to find the pitch out of a spectrum. Unfortunately the strongest component may not be the fundamental one [13].

The time domain and frequency domain methods by themselves are only suitable for very small set of piano songs. This song may contain only monophonic sound (one pitch at a time). The methods are not suitable to chord detection (multiple simultaneous pitches, polyphonic). Problems in time domain approach occur at signals which are not only periodic e.g. signals with noise or polyphonic signals (containing multiple fundamental frequencies simultaneously). Also, the frequency domain approach by itself has a problem with polyphonic detection, but it is possible. Attempts to polyphonic pitch detection are mainly applied in the frequency domain approach [13]. The basic principle include frequency spectrum, which results to amplitudes of peaks. This approach has to be reinforced by several other decision-making and search mechanisms. Many algorithms of these methods perform detection on clean monophonic signal well but failed at noisy signals or polyphonic signals.

Pitch detection is complex problem for monophonic sound, where pitch detection algorithms estimate one pitch at a time. However, there is a need to polyphonic pitch detectors, which can extract multiple pitch at a time or pitches in presence of the noise. This problem is referred to as music transcription or music information retrieval (converting a low-level representation of music into a higher-level representation – MIDI or even music sheets). There are several researches which analyze this problem [1, 6, 9, 12]. Whereas the musical note does not include only the pitch but duration, loudness and timbre [2]. However, detection multiple concurrent pitches [5, 7, 16] is the core of the problem [1]. Further substantial problem is a real-time processing. One way to increase efficiency is to use an iterative principle (e.g. [7]).

Analyses of state of the art in this area with connection with the real-time processing, we found that page turning could be only addressed with the one part of music transcription – note onset detection. This detection based on the control of input data (keystrokes in music sheets) can determine at what position in the song we currently are.

3 Transcription

Music transcription is process of converting musical record into music sheets. This task implies to estimate the pitch, tempo, note onsets, timing of notes, loudness, etc. The task is even more difficult if you are dealing with polyphonic music. If keys on the piano are simultaneously pressed then amplitude in time domain significantly rises. For this reason we focus on the specific problem of music transcription (onset detection) which allows to isolate this change. After our evaluation, we found that accuracy at different input data is not sufficient. Therefore we decide to use control algorithm (PPD) with tracking of played song which used only this method. Both the algorithms operate with some accuracy. After research of the available and implemented methods, we found two methods which are appropriate in terms of efficiency and portability to android device. We analyze two selected and used problems of Music Transcription (spectral flux to onset/keystrokes detection

and summing harmonic amplitudes to PPD). Our method is also appropriate to real-time tracking of song.

3.1 Spectral flux

Spectral flux measures the change in magnitude in each frequency bin [3]. Equation 1 presents summing the positive differences between actual S and the previous frequency LS across all frames, where L is length of spectrum frame.

$$f(t) = \sum_{i=0}^{L} S(i) - LS(i) \tag{1}$$

Keystrokes are determined by peak picking algorithm over $f(t)$. Pre-processing by appropriate threshold function is needed.

3.2 Summing harmonic amplitudes

In [7], there is proposed conceptually simple and computationally efficient fundamental frequency estimator. The estimation is based on summing harmonic amplitudes. It operates in the following steps:

- calculate spectral whitened signal of input signal,
- calculate strength (salience) (Equation 2) of fundamental frequencies candidates as weighted sum of the harmonic amplitudes where, $g(\tau, m)$ is learned by brute force optimization and $f_{t,m}$ is frequency of fundamental frequency candidate.

Spectral whitening suppresses timbre information before actual estimation. Reason of this processing is to make system robust for different input sound sources [7]. It performs by flattening rough spectral energy by inverse filtering [15]. This is done in frequency domain.

$$s(t) = \sum_{m=1}^{M} g(\tau, m)|Y(f_{t,m})| \tag{2}$$

3.3 Estimation of tracking within the song

The problem of tracking within the song only with onset detection is principally with songs characterized by the presence of noise, high tempo, volume level and duration of each individual note. This can result to spurious or omitted keystrokes. There is a need for another control algorithm. We decide to use PPD, which can give clues about type of playing notes. Whereas the problem is the same for both of them, we create solutions which estimate song tracking on the basis of keystrokes with the support of detected notes.

The output of PPD consists of one or simultaneously played notes for each time frame. Length of the frame depends on Fast Fourier Transform window size. Therefore, there are regularly received estimated notes without any information about duration of played notes. Detected peaks from onset detection, thus can give clue about the duration of the notes and also range for note searching. However, peak is not the place where note goes from zero to duration, we add notes' data between the two peaks of some length in addition to currently examined notes. Whereas tracking has to be robust for all durations of song, we empirically found that better results gives the length of $TBTP/3$, where TBTP is the Time Between Two Peaks. So we define note duration time as $TBTP+TBTP/3$.

The algorithm works primarily with onset detection, so we establish decision rules where the detected keystrokes have the largest priority if another check failed. First of all, the algorithm checks

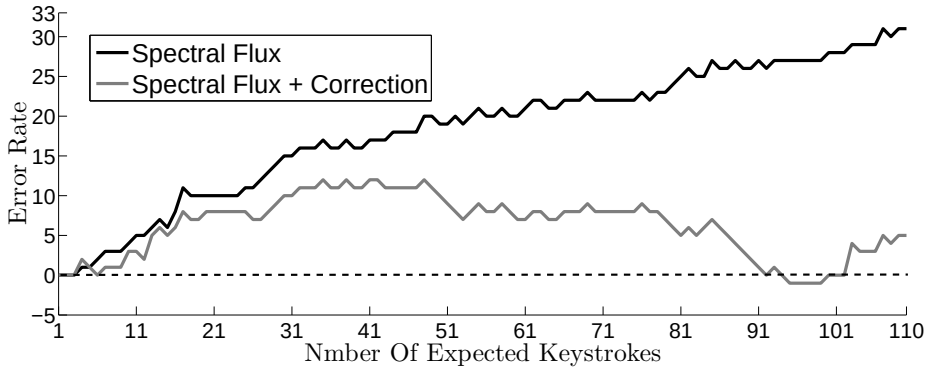

Figure 1. Tracking within the song by spectral flux and correction. Song tempo: 112.

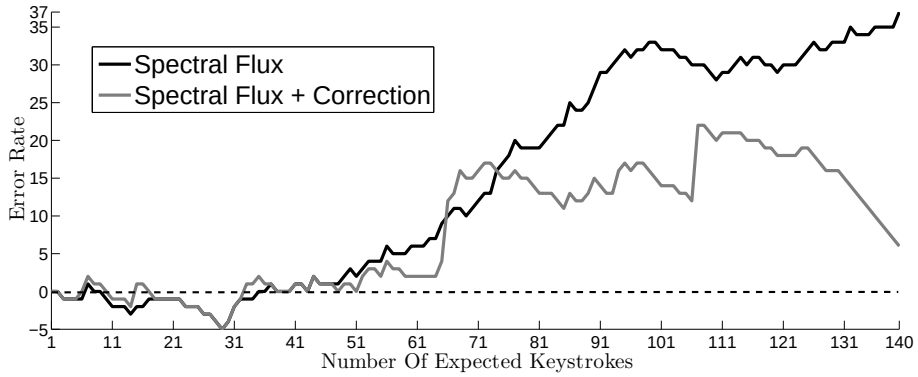

Figure 2. Tracking within the song by spectral flux and correction. Song tempo: 120.

if notes in TBTB are equal with some notes from keystrokes of the input music sheets. If yes, the algorithm considers the keystroke as correct and waits for next keystrokes. If this test failed, we assume a problem with spurious or omitted keystrokes.

We try to eliminate the spurious keystrokes by searching previous keystrokes within the input music sheets. The reason is that if there is a short note duration time, we assumed that there can be an occurrence of previous note, because the note could sounds longer. We try to locate the omitted keystrokes in note duration time by search of keystroke sequence of input music sheets. How many notes in sequence are found, so much are added to the total keystrokes. We also create probabilistic model of comparing the detected notes with input because PPD works in some accuracy. It works on the principle of comparing notes with a note range ($+-0$, $+-1$, $+-2$). We empirically found, if there are results from $+-0$ or $+-1$ in the same test, better results are reported with value which represents their average. This average include number of founded notes. We consider that both results in this range are caused by the inaccuracy of the PPD. We also assume error range $+-2$ in the case of failure of the first two tests of the range. Other results are evaluated on the basis of the results in the presented ranges in sequence.

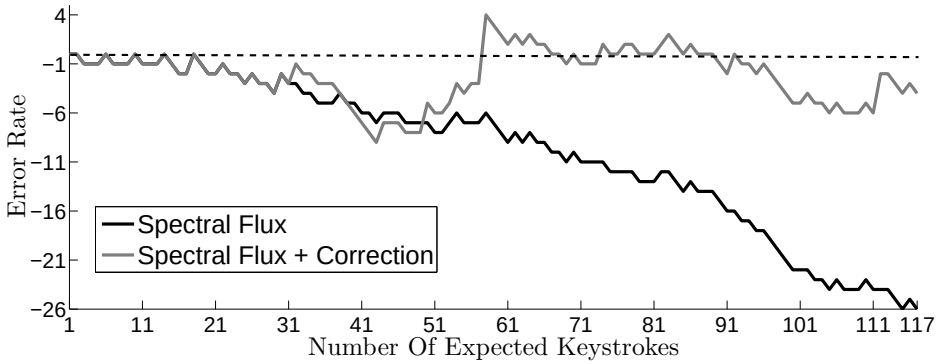

Figure 3. Tracking within the song by spectral flux and correction. Song tempo: 200.

Table 1. Accuracy of the both methods by keystrokes. Spectral flux: TP - correct identified, FP - spurious. Our correction: TP - correct added, FP - false added, TN - correct removed, FN - false removed.

songs		spectral flux		our correction			
tempo	keystrokes	TP	FP	TP	FP	TN	FN
112	110	96.36%	34	0	25	16	20
120	140	77.14%	68	9	30	29	29
200	117	58.11%	22	19	32	1	3

4 Test and evaluation

Two types of input data (song with corresponding music sheets in the form of MusicXML) are used to test our algorithm. We construct MusicXML parser which eliminates keystrokes with relevant notes from music sheets which gives clue of the tracking within the song. We manually annotated the first pages of three songs at each keystroke. Evaluation of the tracking within the song is measured by shift (error rate) against the expected number of keystrokes.

Figures 1, 2 and 3 shows comparison between our algorithm with the method based on the spectral flux only in spurious and omitted keystrokes. Since the spectral flux itself cannot control tracking, each shift has an impact on the final result.

Accuracy of the both method is shown in Table 1. Measurement of our correction include correct added of unidentified keystrokes by spectral flux, false added, correct removed of spurious keystrokes by spectral flux and false removed keystrokes. Our algorithm shows that the wrong identification of spurious and omitted keystrokes brings better results.

5 Discussion and conclusion

We have analyzed algorithms of music transcription and propose algorithm to tracking within the song based on these algorithms. There are additional algorithms related to music transcription which could deal with this problem of tracking within the song, so it is not necessary to perform all the process of music transcription.

Accuracy is influenced by output of both algorithms which still remains to problem of music transcription (robust algorithms which could deal with different types of songs). Tests claim that

synthesis of both algorithms in the despite of their varying accuracy reaches better results. This results are affected by the false detection of spurious and committed keystrokes. In the final analysis, the algorithm provides better results compared to spectral flux itself, what is demonstrated by the tests at three different songs.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., Klapuri, A.: Automatic Music Transcription: Breaking the Glass Ceiling. In: *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012, pp. 379–384.

[2] BYRD, D.B.: Problems of Music Information Retrieval in the Real World. *Computer Science Department Faculty Publication Series*, 2002, p. 4.

[3] Dixon, S.: Onset detection revisited. In: *Proceedings of the 9th International Conference on Digital Audio Effects*, 2006, pp. 133–137.

[4] Gold, B.: Computer Program for Pitch Extraction. *J. Acoust. Soc. Amer.*, 1962, vol. 34, pp. 916–921.

[5] Klapuri, A.P.: Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *Speech and Audio Processing, IEEE Transactions on*, 2003, vol. 11, no. 6, pp. 804–816.

[6] Klapuri, A.: Signal Processing Methods for the Automatic Transcription of Music. Technical report, Tampere University of Technology, 2004.

[7] Klapuri, A.: Multiple fundamental frequency estimation by summing harmonic amplitudes. In: *in ISMIR*, 2006, pp. 216–221.

[8] Noll, A.M.: Cepstrum Pitch Determination. *J. Acoust. Soc. Amer.*, 1967, vol. 41, pp. 293–309.

[9] Paiva, R.P., Mendes, T., Cardoso, A.: Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience, and Melodic Smoothness. *Comput. Music J.*, 2006, vol. 30, no. 4, pp. 80–98.

[10] Phillips, M.S.: A Feature-Based Time-Domain Pitch Tracker. *J. Acoust. Soc. Amer.*, 1985, vol. 77, pp. S9–S10.

[11] Rabiner, L.R., Cheng, M.J., AaronE.Rosenberg, McGonegal, C.A.: A Comparative Performance Study of Several Pitch Detection Algorithms. *IEEE Trans. on ASSP*, 1976, vol. 24, no. 5, pp. 399–418.

[12] Reis, G., de Vega, F.F., Ferreira, A.: Automatic Transcription of Polyphonic Piano Music Using Genetic Algorithms, Adaptive Spectral Envelope Modeling, and Dynamic Noise Level Estimation. *IEEE Transactions on Audio, Speech and Language Processing*, 2012, vol. 20, no. 8, pp. 2313–2328.

[13] Roads, C.: *The Computer Music Tutorial*. MIT Press, Cambridge, MA, USA, 1996.

[14] Schafer, R.W., Rabiner, L.R.: System for Automatic Formant Analysis of Voiced Speech. *Journal of the Acoustical Society of America*, 1970, vol. 47, pp. 634–648.

[15] Tolonen, T., Member, S., Karjalainen, M.: A computationally efficient multipitch analysis model. In: *inria-00350163, version 1 - 6*, 2000, pp. 708–716.

[16] Yeh, C., Roebel, A., Rodet, X.: Multiple Fundamental Frequency Estimation and Polyphony Inference of Polyphonic Music Signals. *Trans. Audio, Speech and Lang. Proc.*, 2010, vol. 18, no. 6, pp. 1116–1126.