Automatic Transcription of Musical Recordings

Anssi Klapuri[1], *Tuomas Virtanen*[1], *Antti Eronen*[1], *Jarno Seppänen*[2]

[1]Tampere University of Technology, P.O.Box 553, FIN-33101 Tampere, Finland
[2]Nokia Research Center, P.O.Box 100, FIN-33721 Tampere, Finland
{klap,tuomasv,eronen}@cs.tut.fi, jarno.seppanen@nokia.com

Abstract

An automatic music transcription system is described which is applicable to the analysis of real-world musical recordings. Earlier presented algorithms are extended with two new methods. The first method suppresses the non-harmonic signal components caused by drums and percussive instruments by applying principles from RASTA spectrum processing. The second method estimates the number of concurrent voices by calculating certain acoustic features in the course of an iterative multipitch estimation system. Accompanying audio demonstrations are at http://www.cs.tut.fi/~klap/iiro/crac2001.

1. Introduction

Transcription of music is defined to be the act of listening to a piece of music and of writing down the musical notation for the sounds that constitute the piece. The scope of this paper is the automatic transcription of the harmonic and melodic parts on real-world musical recordings. Until these days, automatic transcription systems have fallen clearly behind skilled musicians in performance. Some progress has taken place in recent years, however. See [1] for a review of transcription systems.

We have earlier proposed signal processing methods for detecting the beginnings of discrete acoustic events in musical signals [2], and for estimating the multiple pitches of concurrent musical sounds [1]. The multipitch estimator was shown to work reliably in rich polyphonies and to outperform the average of ten trained musicians in musical interval and chord identification tasks. However, the number of concurrent sounds was known in beforehand and noisy operating conditions were not properly addressed. The idea of this paper is to combine the mentioned algorithms and to add what is needed to extend the application area of the system to realistic musical signals.

Overview of the system is shown in Figure 1. The two new algorithms to be proposed in this paper belong to the multipitch estimation stage, and are indicated by rounded boxes in the bottom panel of Fig. 1. Essentially, two modules had to be added to apply the earlier presented multipitch estimation algorithm in real musical signals. The first, noise suppression refers to the suppression of all signal components that do not belong to the harmonic and melodic parts, in practice, drums and percussions. Secondly, number of concurrent voices must be estimated to stop the iterative multipitch estimation algorithm.

2. Overview of earlier presented algorithms

2.1 Onset detection

The onset detector has been originally proposed in [2]. The algorithm employs bandwise processing, building upon the idea that incoming energy at some frequency band indicates the beginning of a physical event that is producing the energy.

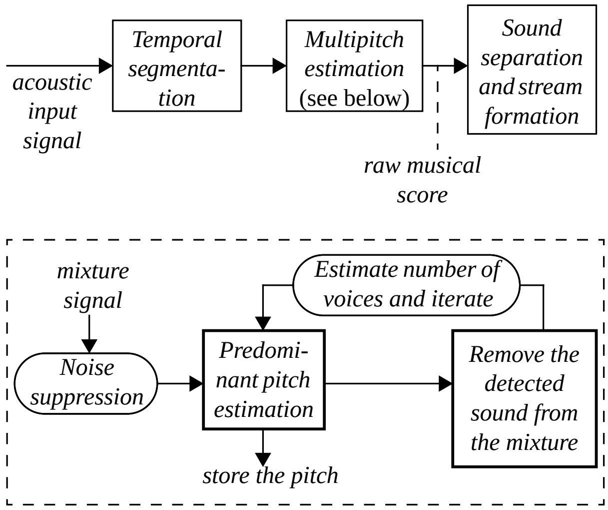

Figure 1: System overview (top), and the parts of the multipitch estimation algorithm (bottom).

A fundamental problem in the design of an onset detection algorithm is distinguishing genuine onsets from gradual changes and modulations that take place during the ringing of a sound. As a solution for this problem, we proposed differentiating the logarithm of the amplitude envelopes at each band. In this case, oscillations in the amplitude envelope do not matter too much after the sound has set on.

2.2 Multipitch estimation

Multipitch estimation forms the core of the transcription system. The algorithm consists of two main parts that are applied in an iterative succession, as illustrated in Fig. 1. The first part, predominant pitch estimation, refers to the crucial stage where the pitch of the most prominent sound is estimated in the interference of other harmonic and noisy sounds. This is achieved by utilizing the harmonic concordance of simultaneous spectral components. In the second part, the spectrum of the detected sound is estimated and linearly subtracted from the mixture. This stage utilizes the fact that the spectral envelopes of real sound sources tend to be continuous. The estimation and subtraction steps are then repeated for the residual signal.

3. Noise suppression

Acoustic noise suppression has been extensively studied in the domain of speech processing. Here the definition of “noise” differs considerably from that in speech processing. Musical recordings practically never have continuous noise that could be estimated over a longer period of time. Instead, non-harmonic parts are due to drums and percussive instruments which are transient-like in nature and short in duration.

Due to the non-stationary nature of the noise, we propose an algorithm which estimates and removes noise independently in each analysis frame. Signal model for a harmonic sound contaminated with additive and convolutive noise is

$$X(k) = S(k)H(k) + N(k) \tag{1}$$

where $X(k)$ is the observed power spectrum of a discrete input signal, and $S(k)$ is the power spectrum of the vibrating system whose fundamental frequency should be measured, for example a guitar string. $S(k)$ has been filtered by $H(k)$, which represents the frequency response of the operating environment and the body of the musical instrument. Suppression of convolutive noise can thus be seen as whitening of the spectra of sounds. $N(k)$ is the power spectrum of unknown additive noise, here represented by all signal components that do not belong to the harmonic part. $N(k)$ cannot be assumed to be stationary.

As a first attempt, we tried to remove additive and convolutive noise in two consecutive stages. First, additive noise was estimated and subtracted in the power spectrum, constraining resulting negative values to a small positive value. Second, logarithm was taken and an estimate of the convolutive noise was subtracted in logarithmic magnitudes. Confirming the reports of earlier authors, two noise reduction systems in a cascade do not produce appropriate results [3]. The overall system tended to work better when either stage was completely bypassed.

Successful noise suppression was achieved by removing both additive and convolutive noise simultaneously, following the lines of RASTA spectral processing [3]. First, the observed power spectrum $X(k)$ is transformed as

$$Y(k) = \ln\{1 + J \times X(k)\}. \tag{2}$$

The value of J acts to scale the input spectrum so that the numerical range for additive noise stays $N(k) \ll 1$, and spectral peaks of the vibrating system $[S(k_p)H(k_p)] \gg 1$, where k_p corresponds to the frequency of a spectral peak p. In this case, additive noise goes through a linear-like transform, whereas spectral peaks, affected by $H(k)$, go through a logarithmic-like transform. Applying a specific spectral subtraction for $Y(k)$ removes additive noise. The spectral envelope of the peak values is flattened by the logarithmic operation itself.

An optimal value of J was found to depend on the level of both the additive noise and the spectral peaks. After experimenting with several different models, the best performance was achieved using the relatively simple expression:

$$J = \alpha\left(\frac{k_1 - k_0}{\sum_{k=k_0}^{k_1} X(k)^{1/3}}\right)^3, \tag{3}$$

which basically calculates the average of the spectrum in the specified frequency range via the cubic root. Indeces k_0 and k_1 correspond to frequencies 50 Hz and 6.0 kHz, respectively, and are determined by the frequency range utilized by the multipitch estimator. Optimal value for α was found to be 1.0.

The noise component $M(k)$ in $Y(k)$ is estimated by calculating a moving average over $Y(k)$ in ERB critical-band frequency scale. More exactly, the magnitude of $M(k)$ for $k=k_i$ is obtained by calculating a Hamming-window weighted average over $Y(k)$ values around k_i, where the width W of the Hamming window depends on the center frequency f corresponding to k_i :

$$W(f) = \beta \times 24.7\left(4.37\frac{f}{1000} + 1\right). \tag{4}$$

In brief, $W(f)$ is β times the ERB critical-band width. Optimal value for β was 4.8. More important than this constant, however, was the observation that estimating noise over ERB frequency scale was clearly advantageous over a linear or the Bark critical-band scale. Among these three, ERB scale is closest to a logarithmic scale, picking an equal amount of spectral fine structure of harmonic sounds over a wide range of fundamental frequency values.

The estimated noise spectrum $M(k)$ is linearly subtracted from $Y(k)$ and resulting negative values are set to zero:

$$Z(k) = \max\{0, Y(k) - M(k)\}.$$

The resulting enhanced spectrum $Z(k)$ is passed to the multipitch estimator, which operates on this enhanced spectrum without returning to the linear magnitude scale.

The accompanying web-page contains audio demonstrations for noise-suppressed musical signals.

4. Estimating the number of concurrent voices

The problem of estimating the number of concurrent voices in music is analogous to voicing detection in speech, with the difference that the output gets integer instead of binary values.

The difficulty of estimating the number of voices is comparable to that of finding the pitch values themselves. Huron has studied musician's ability to identify the number of concurrently sounding voices in polyphonic textures [4]. According to his report, the accuracy in performing the task drops markedly already in four-voice polyphony, where the test subjects underestimated the number of voices present in more than half of the cases. Musical mixtures often blend well enough to virtually bury one or two sounds under the others.

We took a statistical approach to solve the problem. Random mixtures from zero to six concurrent harmonic sounds were generated by allotting sounds from 26 musical instruments of the McGill University Master Samples collection. The mixtures were then contaminated with pink noise or random drum sounds, signal-to-noise ratios (SNR) varying between 23 dB and –2 dB. The drum sounds were taken from a Roland R-8 mk II drum machine.

The iterative multipitch estimation system was run for the generated mixtures, the polyphonies of which were known. Different characteristics of the signal were measured in the course of the iteration – in search for a feature which would indicate the stopping of the iteration after all sounds have been extracted. A number of features were measured, reflecting the level of the extracted sound, residual spectrum, etc.

It turned out to be necessary to perform polyphony estimation using two different models. The first detects voicing, i.e., if there are any harmonic sounds in the input, and the second estimates the number of concurrent voices, if any.

4.1 Voicing detection

Drum sounds turned out to be the biggest problem in voicing detection. Approximately half of the acoustic energy of the sound of bass drums, snares and tom-toms is harmonic, resulting from the drum membrane which vibrates at mode frequencies. This tends to mislead a voicing detector. On the contrary, for pink noise alone, the voicing detector can be designed to work almost perfectly, even though the harmonic sounds them-

Table 1: Voicing detection results.

Problem constraints	extraneous voicing (%)	undetected voicing (%)
93 ms frame, drum noise	1.8	6.1 (2.5)
190 ms frame, drum noise	1.8	1.6 (0.1)

selves vary from the double bass to the transverse flute.

A model for voicing detection in musical signals was found using the procedure described above. A single best feature to indicate voicing was the likelihood L_1 of the sound outputted by the predominant pitch estimator at the first iteration (see Fig. 1). The algorithm for calculating the likelihood in the multipitch estimator has been described in [1]. The best compound feature was derived by combining L_1 with features related to the signal-to-noise ratio of the input signal:

$$V_0 = 4\ln(L_1) + \ln\left(\frac{P_X}{P_M}\right), \tag{5}$$

where P_X is the power of the observed spectrum $X(k)$ in the spectral region between 50 Hz and 6 kHz after the signal has been scaled with J. P_M is the power of the estimated noise spectrum in the same frequency region, calculated by transforming $M(k)$ back to power spectral domain by an inverse transform of Eq. (2). Signal is determined to be voiced when V_0 is greater than a fixed threshold $V_0 > T_{voicing}$.

Table 1 shows simulation results for this model in the presence of drum noise. The results have been averaged over the different polyphonies and over the five different SNRs between 23 dB and –2 dB. Undetected voicing takes place mostly in the –2 dB SNR, which can be noticed from the results in parentheses which have been averaged for noise levels 23 dB…3 dB.

A model which is comparable in accuracy to that of Eq. (5) and does not require the calculation of the algorithm-specific value L_1 can be calculated as

$$V_0' = \ln\left(\frac{P_X P_Z}{P_M}\right), \tag{6}$$

where P_Z is the power of $Z(k)$, the enhanced spectrum.

4.2 Number of concurrent voices

In the case that a section in music has been determined to be voiced, another model is used to control the stopping of the iterative multipitch estimation system, i.e., to estimate the number of sounds to be extracted.

The likelihood L_i of a sound detected by the predominant pitch estimator at iteration i was again a single best feature for controlling the iteration stopping. However, the likelihood values are affected by the SNR, L_i getting smaller in noise. The bias can be explicitly corrected, resulting in the measure

$$V_i = 1.8\ln(L_i) - \ln\left(\frac{P_X}{P_M}\right). \tag{7}$$

As long as the value of V_i stays above a fixed threshold, the iteration is continued and the sound detected at iteration i is included in the output of the multipitch estimator.

The measures of Eqs. (5) and (7) appear as contradictory, but are correct. The SNR-related terms have different roles in these two equations, and thus the different signs. In the estimation of the number of voices, the latter terms corrects for small likelihood values which are due to noise. This is needed to detect soft sounds in noisy polyphonic signals. For voicing detection, there is usually at least one sound prominent enough.

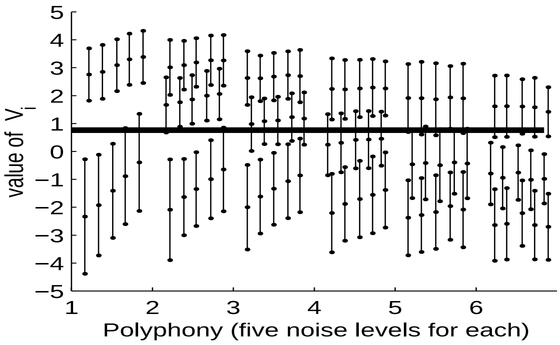

Figure 2: Distribution of the values of V_i in the course of the iteration for polyphonies from 1 to 6, with the drum noise levels –23, –13, –8, –3, 2 dB for each polyphony. Horizontal line shows the threshold where iteration is stopped.

Overestimating the polyphony leads to extraneous notes in the transcription, which has a very disturbing audible effect. Underestimating the polyphony is not very dangerous, since the faintest notes are often not heard out even by human listeners. Overestimation error rate should be very low in all cases.

Figure 2 illustrates the distribution of values of V_i as calculated in a 93 ms frame in the course of the iteration for different polyphonies and drum noise levels. For each polyphony P and noise level there are three lines on top of each other, indicating the value of V_i growing smaller in the course of the iteration. The top lines indicate the mean and standard deviation of V_i for which $i<P$, i.e., V_i values before reaching the actual polyphony. The lines in the middle stand for V_i values for which $i=P$, i.e., for the last legal iteration. The bottom lines stand for V_i at extraneous iterations. Ideally, a thresholding line for V_i should stop the iteration between the stacks of middle and bottom lines. However, to minimize the rate of overestimations, underestimations have to be accepted in rich mixtures.

Table 2 shows the results of the estimation of the number of concurrent voices, averaged over different noise levels.

A model for polyphony estimation which has an acceptable accuracy and does not require the calculation of the algorithm-specific values L_i can be calculated as

$$V_i' = 2\ln(P_i) - \ln\left(\frac{P_X}{P_M}\right), \tag{8}$$

where P_i is the power of the sound detected at iteration i. P_i is obtained by selecting frequency samples from $Z(k)$ from the positions of the harmonic components of the detected sound, transforming them to power spectral domain, and by summing.

5. Sound separation and stream formation

The space permits only a brief mention of the sound separation and stream formation modules. In [5], we have presented a method for the separation of concurrent harmonic sounds. The method is based on a two stage approach, where the described multipitch estimator is applied to find initial sound parameters, and in a second stage, more accurate and time-varying sinusoidal parameters are estimated.

For real musical signals, sound separation is significantly more difficult than for artificial mixtures of clean harmonic

Table 2: Estimation of the number of concurrent voices.

Actual number of voices	Estimated number of voices			
	93 ms frame		190 ms frame	
	drum noise	pink noise	drum noise	pink noise
1	1.1	1.0	1.1	1.0
2	1.9	1.8	2.0	2.0
3	2.6	2.5	2.9	2.8
4	3.1	3.0	3.6	3.4
5	3.5	3.3	4.1	4.0
6	3.6	3.8	4.7	4.4

Table 3: Note error rates in the presence of drum sounds.

Analysis frame size	Polyphony					
	1	2	3	4	5	6
190 ms	6.9	11	14	20	29	39
93 ms	14	20	29	41	51	61

sounds. However, provided that the correct sounds are detected by the multipitch estimator, and that drums do not dominate a musical signal too badly, separation works rather well.

A preliminary attempt towards stream formation from the separated notes was performed by utilizing acoustic features used in musical instrument recognition research [6]. Mel frequency cepstral coefficients, the fundamental frequency, the spectral centroid, and features describing the modulation properties of notes were used to form 17 dimensional feature vectors, which were then k-means clustered. Based on the observations, sream formation according to sources is possible provided that the timbres of the sound sources are different enough, and that the distinctive characteristics do not get lost in the separation process.

6. Simulation results

Table 3 shows the statistical error rate of the overall multipitch estimation system after the noise suppression and polyphony estimation parts were integrated to it. The results have been averaged over three different SNRs: 23 dB, 13 dB, and 3 dB. The test cases were randomly generated from the McGill University samples, pitch restricted between 65 Hz and 2100 Hz. Drum sounds were from the Roland R-8 mk II drum machine.

The error rates in Table 3 have been calculated by summing together inserted, deleted (missing), or erroneously transcribed notes, and dividing the sum by the number of notes in reference. Among the errors, about two thirds were deletions, which is the least disturbing error type. The amount of inserted notes stays around 1 %. The rest are erroneous notes. Noise suppression allows reliable pitch estimation still in 3 dB SNRs.

Together with the onset detector, the system is applicable as such to the transcription of continuous musical recordings. Since exact musical scores were not available for real music, no statistics on the performance are provided. Instead, excerpts from the original signals and synthesized transcriptions for them are available for listening at the accompanying web-page.

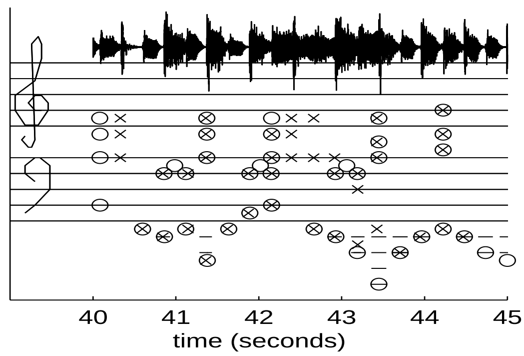

Figure 3: Transcription of a synthesized MIDI-song. Circles denote the original score and crosses the transcription.

Accurate and realistic evaluation of a transcription system is best achieved by transcribing synthesized MIDI-songs. These have the advantage that the exact reference score is available in the MIDI-data. High-quality MIDI-songs are available that are complex enough to simulated real performances. A simulation environment was created which allows reading MIDI-files into Matlab and synchronizing them with an acoustic signal synthesized from the MIDI. Unfortunately, at the time of writing this paper, the transcription system still suffered from certain defects which prevent from publishing error statistics for MIDI-songs. Figure 3 gives an example of a relatively well transcribed song. The piece has regular rock drums, not shown in the score. One defect is that long-duration sounds are detected several times at successive onsets. This results in insertion errors.

7. References

[1] Klapuri, A. P., Virtanen T. O., and Holm, J.–M. (2000). “Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals". In Proc. COST-G6 Conference on Digital Audio Effects, Verona, Italy.

[2] Klapuri, A. (1999). “Sound onset detection by applying psychoacoustic knowledge,” Proc. IEEE International Conf. on Acoust., Speech, and Signal Processing, Phoenix, Arizona, 1999.

[3] Hermansky, H., Morgan, N., Hirsch, H.-G. (1993). “Recognition of speech in additive and convolutiove noise based on RASTA spectral processing,” IEEE International conference on Acoustics, Speech, and Signal Processing, Minneapolis, Minnesota, 1993.

[4] Huron, D. (1989). “Voice Denumerability in Polyphonic Music of Homogeneous Timbres,” Music Perception, Summer 1989, Vol. 6, No. 4, 361–382.

[5] Virtanen, T., Klapuri, A. (2001). “Separation of harmonic sound sources using multipitch analysis and iterative parameter estimation,” Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York.

[6] Eronen, A. (2001). “Comparison of features for musical instrument recognition,” Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York.