Low Level Descriptors for Automatic Violin Transcription

Alex Loscos
Universitat Pompeu Fabra / National University of Singapore
Music Technology Group / School of Computing
aloscos@iua.upf.edu / loscos@comp.nus.edu.sg

Ye Wang
National University of Singapore
School of Computing
Singapore 117543
wangye@comp.nus.edu.sg

Wei Jie Jonathan Boo
National University of Singapore
School of Computing
Singapore 117543
booweiji@comp.nus.edu.sg

Abstract

On top of previous work in automatic violin transcription we present a set of straight forward low level descriptors for assisting the transcription techniques and saving computational cost. Proposed descriptors have been tested against a database of 1500 violin notes and double stops.

Keywords: Violin, Automatic Transcription.

1. Introduction

Automatic transcription is a problem that has been addressed using many different approaches [1]. Most of these tackle the problem from an instrument-free perspective such as 'music transcription' or 'monophonic and polyphonic transcription'. And among the ones that specialize on specific sections (drums and percussion) or instruments (piano) very few focus on violin [2][3][4].

In this context, this paper aims a step forward in the implementation of a clear-cut violin transcription system first described in [5], originally thought to be used for distant education and self learning and evaluation. The paper introduces a set of low level descriptors by which the system can improve its performance and adapt the complexity of the analysis algorithms that are being applied. The general block diagram of the system is represented in figure 1.

Following sections introduce improvements in the pitch estimation; present first derivative zero crossing descriptor and modulation descriptor for upper octave polyphony detection; and inharmonic descriptor for any other duo-phony detection. So once the pitch analysis receive a note, stability descriptor decides which frames to use for the pitch estimation, and inharmonic descriptor decides whether pitch estimation algorithm deals with a single note or a double-stop regions. For those cases in which monophonic pitch analysis has been applied, the upper octave descriptors, one of them using the already estimated pitch, decide whether the upper octave note has to be given as transcription output as well.

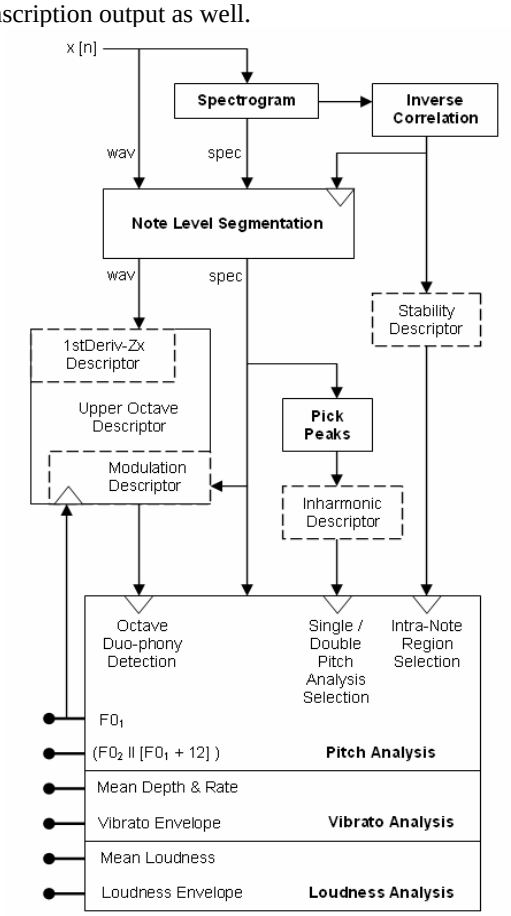

Figure 1. General block diagram of the automatic violin transcription system where dotted lines represent descriptors, triangles represent controls and rounded-end lines represent outputs

2. Note Level Segmentation

Note Level segmentation uses implementation from previous system [5] based on the autocorrelation of the so called Note Spectrogram. While this approach is efficient sorting out monophonic pitch changes, it sometimes lacks of resolution for detecting timbre modulations, amplitude

modulations, and pitch modulations. This, far from being a drawback, allow the note level segmentation to work free of such kind of modulations, which enrich the sound but do not define the note itself. However, specific cases such as note repetitions, fast performance, or deep modulations require from additional processing.

3. Pitch Analysis

Previous system [5] already confronted the problem of octave errors by means of adding a compression term ruled by parameter *a* (set to 5 in [5]) in the summation of the Semitone Band Spectrum (SBS) *p(w)* as formulated in equation:

$$A(w) = \sum_{k=1}^{5} \min\left(p(k \cdot w), a \cdot p(w)\right) \quad (1)$$

This compressed addition has been proven useful for getting rid of lower octave errors. When *w* is set to be $F_0/2$ (being F_0 the pitch), we add the real fundamental and harmonics up to a certain proportional value. If no peak is at $F_0/2$ no relevant energy will is added to *A(w)*. However, as we discuss in section 3.1.2, this technique does not get rid of upper octave errors

3.1 Improved Pitch Estimation

A couple of generic modifications (both for monophonic and polyphonic pitch estimation) are introduced: steady state detection (lightens computational cost and avoids transients), and octave error pruning.

3.1.1 Stability Descriptor

For each of the note regions outcome by the note-level segmentation, a stability descriptor is computed so that pitch estimation is performed using only the most stable sub segment of the note. This selection is done by means of the inverse correlation descriptor, computed as ion [5].

3.1.2 Upper Octave Pitch Error

Pitch estimation of monophonic signals is a problem that can be considered solved. Only upper octave errors take place when analyzing the monophonic note samples using the formulation presented in [5]. These errors take offset values of 12 or 19 semitones which correspond to the second and the third harmonic respectively.

In order to solve such errors a new term is added in the summation of the harmonics in the SBS. This term punishes the energy in the sub-harmonics frequencies using the following expression:

$$A(w) = \sum_{k=1}^{5} \min\left(p(k \cdot w), a \cdot p(w)\right) + b \cdot \left(\sum_{k=1}^{4} p\left(k \cdot w - \frac{w}{2}\right) + \sum_{k=1}^{3} p\left(k \cdot w - \frac{w}{3} - \frac{2 \cdot w}{3}\right) \right) \quad (2)$$

After running several experiments, ß was set to 3. With such value we achieved error-free pitch estimation for monophonic notes.

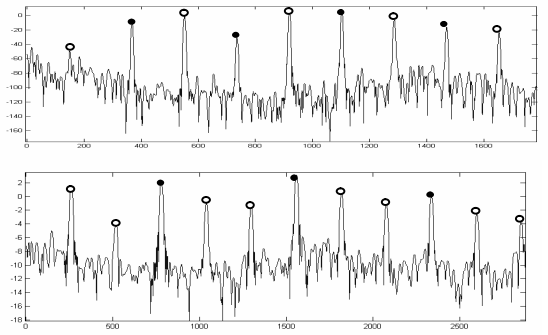

Figure 2. Local view of spectrums for typical pitch errors: B3 second harmonic confusion error (up), and F4 third harmonic confusion error (down). Black dots show mistaken fundamental and harmonics, empty dots show the rest of original harmonic peaks.

3.2 One Octave Duo-phony Detection

Previous system did not consider polyphonies in which one of the notes was an octave higher or lower from the other. The difficulty of pitch detection in such cases is that there is 100% overlap of harmonics, i.e. the harmonic spectra of the higher octave note hides under the lower octave note harmonics. In order to take into consideration such cases, we propose to include a detector in charge of resolving whether the note being considered was played together with its upper octave note or not.

Our octave duo-phony detector is based on the analysis of the magnitude spectrum modulation around the even harmonics (*2*k*F_0 for k=1,2,3,..*), and the zero crossing factor of the first derivative of the low-pass filtered waveform.

3.2.1 Modulation Descriptor

Since the violin is not a perfect tuned instrument [1], the assumption is that whenever we have an octave distance duo-phony, even harmonics will suffer from amplitude and frequency modulations because of the frequency juxtaposition. The modulation descriptor is formulated as the mean value:

$$Md = \frac{\overline{\sum \left|\Delta_t FSpectrogram\right| \cdot AvSpectrogram}}{\sum AvSpectrogram} \quad (3)$$

where AvSpectogram is the spectrogram averaged along time and *FSpectrogram* is the spectrogram of the input signal filtered by the adaptative FIR comb filter which has zeros placed over the fundamental frequency and odd harmonics, as shown in figure 3.

$$y(n) = \frac{1}{2} \cdot \left(x(n) + x\left(n - \left[\frac{F_s}{2 \cdot F_0} \right] \right) \right) \quad (4)$$

being F_s the sampling frequency and F_0 the estimated pitch.

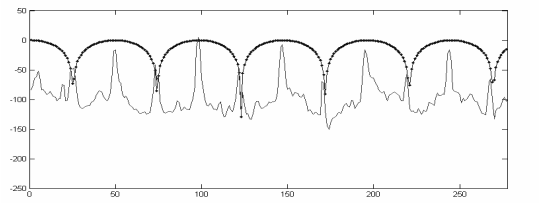

Figure 3. Local view (lower spectra) of the F_0 dependent comb filter (dotted line) and the resulting filtered violin average spectra (log magnitude versus frequency index)

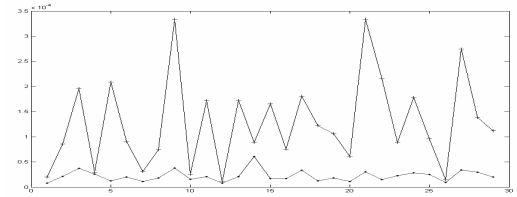

Figure 4. Amplitude/Frequency modulation along notes (A#3, A#4, A#5, A3, A4, A5, B3, B4, B5, C#4, C#5, C4, C5, D#4, D#5, D4, D5, E4, E5, F#4, F#5, F4, F5, G#4, G#5, G3, G3, G4, G5) for duo-phonic (+), and solo lower note (.)

Figure 4 shows the values of our modulation descriptor along our octave duo-phony recordings. In average terms the modulation descriptor obtained in the octave duo-phonic notes (~1.3e-4) are six times the values obtained for the solo notes (~2.2e-5). Notice modulation parameter has a strong dependence on performance. Note 26 and 27 refer to the same note (G3) but performed in a completely different way; in 26 the upper octave is extremely subtle and very well tuned.

3.2.2 First Derivative Zero Crossing Descriptor

The zero crossing factor of the first derivative of the waveform counts the number of times a signal changes from decreasing to increasing and vice versa. The assumption is that by previously low pass filtering the waveform to get rid of the highest frequency components, in those cases in which the signal is the summation of two components at an octave distance, the descriptor will be proportional to the highest pitch.

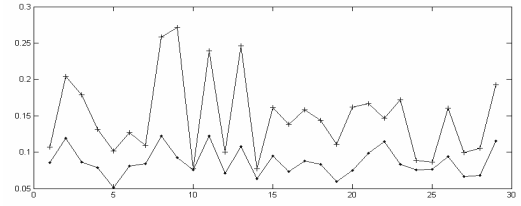

Figure 5. Updown parameter obtained from 29 files (A#3, A#4, A#5, A3, A4, A5, B3, B4, B5, C#4, C#5, C4, C5, D#4, D#5, D4, D5, E4, E5, F#4, F#5, F4, F5, G#4, G#5, G3, G3, G4, G5) for duo-phonic (+), and lower (.) notes.

Results shown in figure 5 use a 10 point average filter. The mean descriptor value obtained in the octave duo-phonic notes (~0.15) is around twice the mean value obtained for the solo notes (~0.08). Lowest values such as the one obtained for note 10 (C#4) most of the times are due to an extremely softly upper octave note, being this note only distinguishable at the release or transition.

3.3 Duo-phony Detection for Double Pitch Estimation

The harmonic descriptor is in charge on guiding the pitch estimation process by telling if the input note is monophonic and thus, it can use straightforward pitch estimation techniques or it is polyphonic and thus it requires additional processing for the estimation of the two notes. Obviously, octave distance duo-phonies should be here detected as monophonic.

The inharmonic descriptor is based on the spectrum peaks distribution along frequency. While most harmonic related descriptors base their analysis in a prior knowledge of the pitch, our method is blind-pitch. The descriptor picks first eight most prominent magnitude spectral peaks and measures the divisibility among frequency distances defined by all possible different pairs of them.

The spectral peak detection uses a modification of *PickPeaks* procedure from [6] in which only peaks above a lower frequency boundary (which is set to the lowest possible violin pitch) are considered, and all peaks below an adaptative noise threshold are discarded. Being *PeakFreq* the vector containing sorted frequency positions of the peak, the descriptor can be formulated as:

$$InHd = \overline{\sum_{i=1}^{8}\sum_{j=i}^{8}\left| res\left(\frac{\Delta_f\left(PeakFreq\{i\}\right)}{\Delta_f\left(PeakFreq\{j\}\right)}\right)\right|} \qquad (5)$$

$$\text{where} \quad res(x/y) = y\cdot[x/y] - x$$

Most non-duo-phony notes with high inharmonic score happen for the highest violin notes, when bow noise becomes more significant, spectral peaks become more distant among them and some parasite peaks are gathered mistakenly.

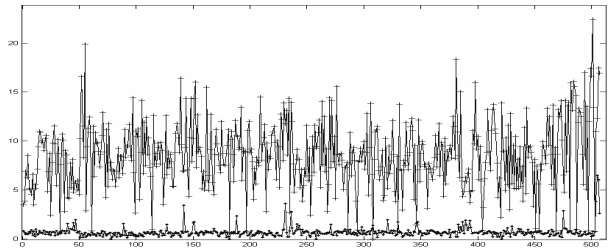

Figure 6. Inharmonic descriptor obtained from 478 files for duo-phonic (+), and monophonic (.) notes.

Pitch estimation technique presented in section 3.1 does not straightly fit into duo-phonic pitch analysis. Formulations such as (1) and specially (2) do not make sense anymore since $F_0/2$ and $F_0/3$ might conflict with

second note harmonics. Better results can be achieved by trying to get two pitches at the time instead of two step analyses (analysis, subtraction, and analysis) in a very similar way as presented by [7].

4. Vibrato Analysis

In the violin, vibrato is produced by a periodic rolling motion of a finger on the string, which produces a frequency modulation. Because of instrument body resonance, the frequency modulation (FM) generates amplitude modulations (AM) in each partial [8]. FM and AM nearly always coexist in all musical instruments in a way that it is impossible to have frequency vibrato without amplitude vibrato but not vice versa [8]. In the case of violin vibrato, AM seems to be perceptually more relevant than FM [9]. One may naturally conclude AM to be the natural feature choice for vibrato detection. However, experiments show there is strong correlation between fundamental's FM and partial's FM while no such correlation appears for AM.

Because of all previous considerations, the vibrato analysis proposed for our automatic transcription focuses on FM to determine vibrato presence and FM and AM to characterize its manifestations. Our current vibrato analyzer is implemented on top of Liu's [10] proposal, where the so called Time-Varying Amplitude (TVA) and the Time-Varying Frequency (TVF) for the jth harmonic at the mth time frame is calculated by formulas:

$$TVA_n^m = \sqrt{\sum_j \left|S^m\left(f_j\right)\right|^2} \qquad (6)$$

$$TVF_n^m = \sum_j f_j \cdot \left|S^m\left(f_j\right)\right|^2 \Big/ \sum_j \left|S^m\left(f_j\right)\right|^2 \qquad (7)$$

where in our adaptation, j takes values between 0 and 4, $S^m(f)$ is the spectrum of frame m, and fj (being f_0 the fundamental) covers, for every index j, a range of 100 cents centered around the j^{th} harmonic frequency.

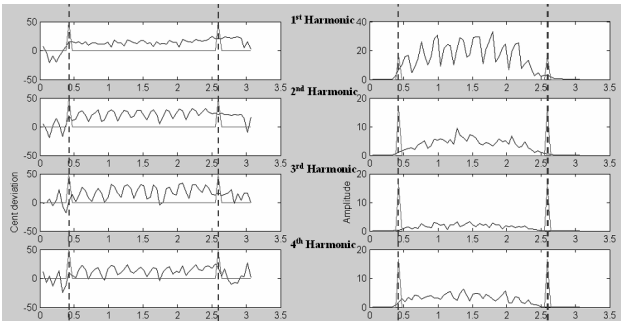

Figure 7. *TVF* (left column) and *TVA* (right column) for the first four harmonics of an A4, performed with vibrato. Dashed lines bound considered vibrato regions.

5. Concluding Remark and Future Work

This paper has presented the current state of an ongoing research project. For our intended application, we have studied the problem of note-level segmentation, pitch estimation, and vibrato analysis. Experience tells us reliable note-level segmentation based solely on audio signal is a very challenging goal to achieve. Preliminary results show that it is possible to improve note-level segmentation with the help of visual cues from associated video clips. This approach is currently under progress. We also plan to include dynamics attribute extraction in future releases since its importance in the educational context, especially for advanced students. Regarding pitch estimation, although accuracy has been improved due to our modifications on previous system, we believe it should allow a non fixed scale analysis. This is a very demanding requirement for amateur students that are getting to learn violin and can perform significantly out of tune.

References

[1] Klapuri, A, "Automatic music transcription as we know it today," Journal of New Music Research, Vol. 33, No. 3, pp. 269-282, Sep. 2004.

[2] J. Yin, Y. Wang, D. Hsu, "Digital Violin Tutor: An Integrated System for Beginning Violin Learners" ACM Multimedia, Hilton, Singapore, 2005.

[3] Robert Scott Wilson, "First Steps Towards Violin Performance Extraction using Genetic Programming", In John R. Koza editor, Genetic Algorithms and Genetic Programming at Standford 2002, pages 253-262, Stanford, California, 2002.

[4] A. Krishnaswamy and J. O. Smith, "Inferring control inputs to an acoustic violin from audiospectra", in Proceedings of the International Conference on Multimedia Engineering, New York,2003, IEEE Press.

[5] Boo, J. Wang, Y. Loscos, A., "A Violin Music Transcriber for Personalized Learning", in Proceedings of the ICME06 Conference, Toronto, Canada, 2006.

[6] Udo Zölzer et al. "DAFX - Digital Audio Effects', ISBN: 0-471-49078-4, John Wiley & Sons, 2002, Chapter 11. Available: http://www2.hsu-hh.de/ant/dafx2002/DAFX_Book_Page/matlab.html

[7] Takuya Fujishama, "Real-time chord recognition of musical sound: A system using Common Lisp Music", in Proceedings of the International Computer Music Conference, Beijing, 1999

[8] Rossing, T.D., N.H. Fletcher, "The Physics of Musical Instruments", Springer, 2nd edition, 1998.

[9] Järveläinen, H., "Perception-based control of vibrato parameters in string instrument synthesis", in Proc. International Computer Music Conference, Sweden, 2002

[10] Ning, Liu. "Vibrato analysis and synthesis for violin: an approach by using energy density spectrum analysis", internal report, National University of Singapore, 2005