COMPARISON OF PITCH TRACKERS FOR REAL-TIME GUITAR EFFECTS

Adrian von dem Knesebeck,Udo Zölzer

Dept. of Signal Processing and Communications,
Helmut Schmidt University
Hamburg, Germany
knesebeck@hsu-hh.de

ABSTRACT

A comparison of various pitch tracking algorithms is presented and the suitability to use these pitch trackers for real-time guitar signal pitch tracking is investigated. The pitch tracking algorithms are described and the performance regarding latency and accuracy is evaluated.

1. INTRODUCTION

The task of pitch detection and tracking has often been discussed and a lot of approaches to fulfill this task have been presented in the past decades. This paper is not intended to propose yet another pitch tracking algorithm, but will compare existing algorithms and discuss the usability for real-time pitch tracking for digital audio effects.

The pitch can be used to control some kind of effect parameter or to do processing on the input signal itself. Both cases demand different performance properties. The adjustment of audio effect parameters is not as time critical as the processing of the audio signal, because any interruption of the audio signal is directly perceivable as annoying click or artifact.

We want to track the pitch of electric guitar audio signals. The desired application is to apply pitch transposing effects, e.g. to perform pitch correction or to enhance the signal with harmony signals according to the musical context. This application demands a very low latency of the detector to enable live performance and a low computational cost to meet the requirements for real-time processing. The paper is structured as follows: In section 2 we present the investigated pitch tracking approaches. First we describe the general idea of the autocorrelation based time domain pitch trackers. This basic approach is not used to compete with the other approaches described, but gives a basic introduction to the time domain approaches examined. One frequency domain approach is included in the comparison. The different approaches are compared regarding latency and detection robustness using a set of test data in section 3. Section 5 discusses the results and concludes the paper.

2. PITCH TRACKING ALGORITHMS

The detection of a pitch of a monophonic sound can be regarded as a problem of fundamental frequency detection for most instruments. The sounds that we are interested to track in this paper are dry electric guitar signals generated by playing single notes, resulting in monophonic sounds.

2.1. Autocorrelation

The probably most famous function for pitch detection is the autocorrelation function (ACF). The ACF is the sum of multiplications of a time window with its shifted version and reveals the periodicity of the signal. The ACF sequence for a block of length N with lag l is defined as

$$r(l) = \sum_{n=0}^{N-1} x(n)x(n-l). \tag{1}$$

The ACF shows peaks at time lags of high correlation, i.e. for sinusoidal signals the peaks show up at the periods of the contained signals. The highest peak of the ACF occurs when $l = 0$, hence a minimum lag l_{min}, which corresponds to the highest detectable frequency, has to be defined. Assuming the fundamental frequency is the most prominent frequency, the fundamental period can be determined by picking the maximum peak of the ACF, which has a lag greater than l_{min}:

$$l_{max} = \arg\left(\max\left(r\left(l\right)\right)\right), \qquad \text{with} \quad l > l_{min}. \tag{2}$$

The fundamental frequency then is given by the reciprocal of the period,

$$f_0 = \frac{1}{l_{max}}. \tag{3}$$

Since we obtain discrete lag values for the discrete ACF, the resolution of the detected frequencies is given by the frequency error factor $\alpha(f_0)$ determined by the ratio of the exact frequency f_0 and the detected "discrete" frequency $\tilde{f}_0$, [1]. The error factor depends on the fundamental frequency f_0 and the sampling frequency f_s and is defined by

$$\alpha(f_0) = \frac{f_0}{\tilde{f}_0} = 1 + 0.5\frac{f_0}{f_s}. \tag{4}$$

A computational efficient way to calculate the ACF is to perform inverse Fourier transform of the power spectrum [2]. The ACF method is prone to octave errors, because the peaks of the ACF are periodically repeated. Therefore the range of observed lags has to be carefully chosen and further processing is beneficial to increase the robustness of detection. A method taking benefit of additional processing steps is described in section 2.3.

2.2. Long Term Prediction

The pitch period can also be determined by a long term prediction (LTP) approach which makes use of the autocorrelation as

described in [1]. The long term prediction error of a one-tap FIR filter with a delay line of M samples is given by,

$$d(n) = x(n) - b_0 \cdot x(n - M). \tag{5}$$

The optimal filter coefficient, which minimizes the energy of $d(n)$ over one block of length N is calculated as

$$b_0 = \frac{\tilde{r}_{xx}(M)}{r_{xx0}(M)}, \tag{6}$$

with the exact autocorrelation, which uses samples preceding the considered block,

$$\tilde{r}_{xx}(l) = \sum_{n=0}^{N-1} x(n)x(n-l) \tag{7}$$

and the energy of a delayed block by l samples given by

$$r_{xx0}(l) = \sum_{n=0}^{N-1} x^2(n-l). \tag{8}$$

Using the normalized autocorrelation

$$r_{xx,norm}(l) = \frac{\tilde{r}_{xx}(l)^2}{r_{xx0}(l)}, \tag{9}$$

the energy depending on M can be expressed as

$$E_d = \sum_{n=0}^{N-1} \big(x^2(n) - r_{xx,norm}(M)\big). \tag{10}$$

To minimize this energy the lag $l = M$ has to be found which maximizes $r_{xx,norm}(l)$.

The determination of the pitch lags is done in three steps. First the local maxima in $r_{xx,norm}(l)$ are searched. Since all values of $r_{xx,norm}(l)$ are positive and there are also local maxima where $\tilde{r}_{xx}(l)$ has minima, the next step is to select only the local maxima where $\tilde{r}_{xx}(l)$ also has positive values. These selected maxima are pitch lag candidates. In the third step the coefficient b_0 is used. The value of b_0 is close to 1 for voiced sounds and close to zero for unvoiced sounds. Hence the values of $b_0(l)$ above a certain threshold close to 1 can be considered as pitch candidates. So the overall pitch lag is chosen to be the pitch candidate with the lowest lag value which has a common pitch candidate at the b_0 candidates and the selected maxima candidates.

2.3. PRAAT

An enhanced ACF approach is implemented in the PRAAT speech analysis tool as described in [3]. The enhancement involves the following steps. The robustness regarding octave errors has been improved by eliminating the influence of the window function on the ACF. A window $w(n)$ is applied to the time frame $x(n)$ resulting in a signal $a(n) = x(n) \cdot w(n)$. The Gaussian window was found to be the window of choice. The ACF of the windowed signal $a(n)$ is calculated returning $r_a(l)$. The ACF of the window itself is calculated as $r_w(l)$. An approximation of the ACF of $x(n)$ can now be calculated as

$$r_x(l) \approx \frac{r_a(l)}{r_w(l)}. \tag{11}$$

Another enhancement is to increase the frequency resolution by up sampling and performing sinc interpolation on the autocorrelation signal. For each time frame a number of pitch candidates is determined. The one pitch candidate with highest strength is chosen as locally best candidate. The strength can be simplified described as the weighted absolute peak values of the ACF above a certain threshold. For a detailed description of the pitch candidate selection and the tuning parameters of the strength values we refer to [3].

The PRAAT tool includes an optional post-processing step. The "correct" pitch candidate of each frame is selected by finding the globally least cost path through the frames using the Viterbi algorithm. High frequency deviations like octave jumps result in a higher cost. For the comparison done in this paper this post-processing step was omitted to yield results valid for a real-time application of this approach.

2.4. YIN

Another way to determine the periodicity of a signal is to calculate the sum of differences of a time frame with its shifted version analogous to the ACF. The average magnitude difference function (AMDF) [4] is defined as

$$d(l) = \frac{1}{N} \sum_{n=0}^{N-1} |x(n) - x(n-l)|. \tag{12}$$

The function properties are similar to the ACF, but the AMDF shows dips at the lags of high correlation instead of peaks like the ACF does. De Cheveigné [5] defined the difference function as sum of squared differences

$$d_t(l) = \sum_{n=1}^{N-1} \big(x(n) - x(n+l)\big)^2. \tag{13}$$

The ACF and the difference function both are sensitive to amplitude changes of the time signal. An increasing amplitude of the time signal leads to higher peaks at later lags for the ACF and lower dips for the difference function respectively. To avoid this an cumulative normalization is applied to average the current lag value with the previous values. The result is the normalized mean difference function (NMDF) defined as

$$d'_t(l) = \begin{cases} 1, & l = 0 \\ \dfrac{d_t(l)}{\frac{1}{l}\sum_{n=1}^{l} d_t(n)}, & \text{else.} \end{cases} \tag{14}$$

The NMDF starts at a value of 1 and drops below 1 only where the current lag value is below the average of all previous lags. The minimum of the parabola is used as the refined lag value. This allows to define a threshold, which a local minimum $d'_t(l)$ has to fall below in order to consider it as a valid pitch candidate. To increase the frequency resolution of the YIN algorithm the local minima of $d'_t(l)$ are refined by parabolic interpolation with their neighboring lag values.

2.5. Spectral Peak Picking

The fundamental frequency of a sound can obviously be determined in frequency domain, i.e. in the FFT spectrum. The peak of the magnitude response with the lowest frequency is picked and

can be regarded as the fundamental frequency of the sound. There are some drawbacks with this straight forward frequency domain approach. First of all the FFT resolution is constrained with the length of the FFT frame. A N-point FFT has a resolution of

$$\Delta f = \frac{f_s}{N}. \tag{15}$$

Since the FFT bins are linearly distributed the resolution at low frequencies is rather poor, e.g. a 1024-point FFT has a resolution of $\Delta f \approx 43$Hz at a sampling rate of $f_s = 44.1$kHz. This is definitely insufficient for guitar pitch determination. In standard tuning the pitch of the empty lower E string of a guitar has a fundamental frequency of 84.4Hz and the pitch of the F on the first fret has a fundamental frequency of 87.3Hz. One way to increase the frequency resolution of the FFT is to make use of the phase information as described in [1]. Each FFT bin by definition represents one harmonic component

$$x_h(n) = cos(\Omega_0 n + \varphi_0) = cos(\phi(n)), \tag{16}$$

with $\Omega_0 = k_0 \frac{2\pi}{N}$. Therefore the fundamental frequency is the derivative of the cosine argument ϕ by n,

$$\Omega_0 = \frac{d\phi(n)}{dn}. \tag{17}$$

This derivation can be approximated by using the phase difference of two FFTs with a hop size of R samples,

$$\hat{\Omega}_0 = \frac{\Delta\phi(n)}{R}. \tag{18}$$

The frequency resolution of a pitch candidate at frequency k_0 can be enhanced by calculating the expected phase φ_{2t} after a progression of R samples as

$$\varphi_{2t} = \varphi_1 + \frac{2\pi}{N} k_0 R, \tag{19}$$

with φ_1 being the phase angle at bin k_0 of the first FFT. The real phase progression after R samples is determined as the phase angle φ_2 at bin k_0 of the second FFT. The phase error is given by $\varphi_{2err} = \varphi_2 - \varphi_{2t}$. The real unwrapped phase is calculated as sum of the expected phase and the phase error,

$$\varphi_{2u} = \varphi_{2t} + \varphi_{2err}. \tag{20}$$

This allows to calculate the corrected frequency of the pitch candidate as,

$$\hat{f}_0 = \frac{1}{2\pi}\hat{\Omega}_0 f_s = \frac{1}{2\pi}\frac{\varphi_{2u} - \varphi_1}{R} f_s. \tag{21}$$

The pitch candidates are selected by picking the peaks of the FFT spectrum which have a magnitude above a certain threshold. Since a pitched guitar string produces several harmonics, there will be more than one pitch candidate. So the estimate of the fundamental frequency of the examined frame is set to be the lowest frequency pitch candidate.

3. PITCH TRACKER PERFORMANCE

To evaluate and compare the performance of the pitch trackers it is necessary to define the performance criterions. The most important criterion is the accuracy of the detected pitch. Since we are looking for a real-time pitch tracker, the latency as well as the computational complexity also play an important role.

Table 1: Mean absolute error of detected pitches in Hz.

Pitch Tracker	LTP	PRAAT	YIN	FFT
Test 1	0.3341	0.1080	0.0495	0.3929
Test 2	0.5059	0.2233	0.0836	0.1045

Table 2: Latency of the pitch tracker algorithms in ms.

Pitch Tracker	LTP	PRAAT	YIN	FFT
with post processing	69.7	-	27.4	69.7
w/o post processing	23.2	40	13.4	23.2

3.1. Test Signals

The comparison of the pitch trackers described in section 2 is targeted on the performance with guitar signals. Therefore the set of test signals has to be chosen to cover the common guitar playing techniques. For the evaluation of the accuracy we decided to use synthetic signals. This way the instantaneous frequencies as well as the onset times are exactly known, which allows to define a deviation error. As test signal 1 (Test 1) we used an ascending C major scale ranging from 130.8Hz to 261.6Hz and as test signal 2 (Test 2) a vibrato signal with a fundamental frequency of 293.7Hz, which is a high D, with a vibrato frequency of 6Hz. Since we are interested in tracking real guitar signals we also recorded an electric guitar. The dry guitar signal was directly fed into the sound card. We do not know exactly the frequencies that occur while playing the guitar, because pressing the strings on the frets of the guitar always results in some degree of detuning. Also the exact onsets of the played notes are unknown. Therefore we did not use the real guitar signals for the accuracy estimation but for a qualitative comparison. As test licks we used the C major scale in the same frequency range as the synthetic signal.

3.2. Accuracy

We evaluate the accuracy in terms of the mean absolute frequency deviation in Hertz. Since the pitch detectors sometimes return a pitch value where no signal is and sometimes do not detect a pitch where there is a signal, we decided to evaluate only those parts where a pitch was detected and a signal was present. This means we evaluate the accuracy of true pitch detections to enable a comparison of the pitch frequency estimation accuracy of the different pitch trackers. Table 1 shows the results for the two test signals Test 1 and Test 2.

3.3. Latency

All investigated pitch trackers are frame-based and therefore introduce a latency depending on the observation frame length. The minimum observation frame length is constrained by the period time of the lowest detectable frequency. For this comparison the minimum frequency was set to 75Hz which leads to a minimum period time of 13.3ms. The pitch trackers all introduce additional latency depending on the approach. The LTP, YIN and FFT algorithms include post-processing steps which use neighboring pitch estimates to increase the stability of the detection. The PRAAT algorithm also offers a post-processing step, but this processes the whole set of detected pitches as described in section 2.3. Therefore

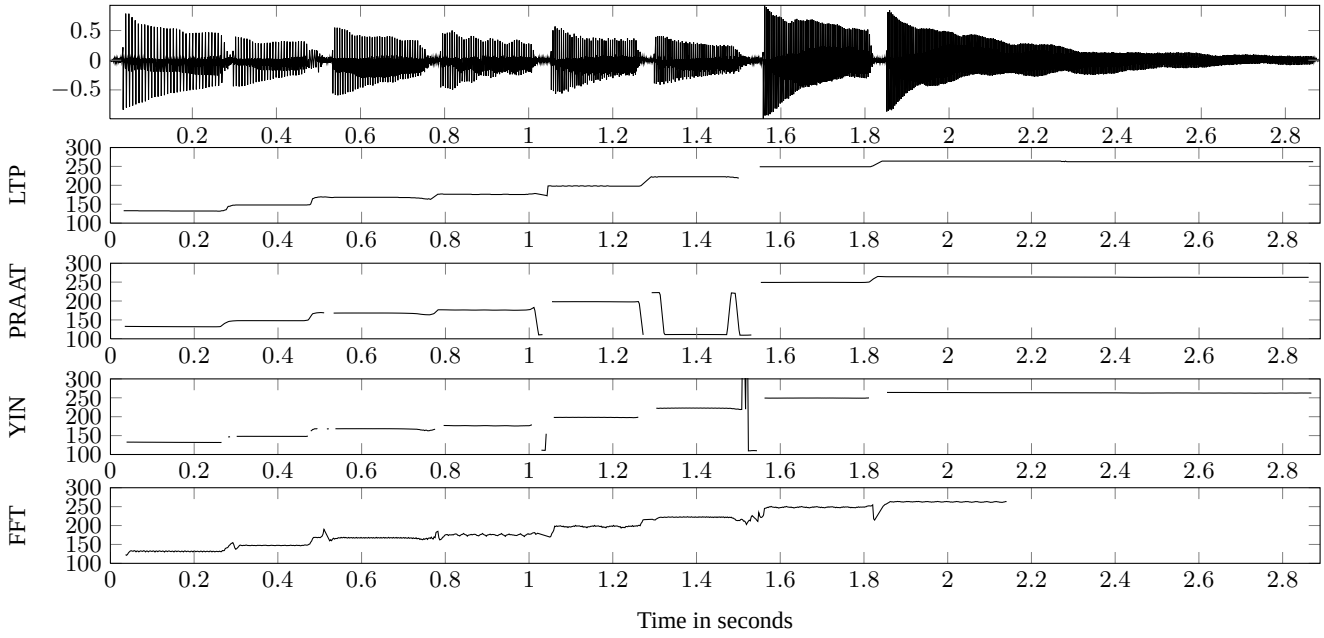

Figure 1: Pitch tracking results with recorded guitar signal. The top plot shows the waveform of the guitar signal. other plots show the detected pitches with the frequency on the y-axis.

it is not suitable for real-time application. The latencies achieved with and without the post-processing step are given in table 2.

4. DISCUSSION

The results of the accuracy evaluation show that for synthetic signals all investigated pitch trackers return suitable pitches. The YIN algorithm overall performs best with a mean error of below 0.01Hz and a latency of 27.4ms. The PRAAT algorithm has some difficulties to keep track of the vibrato signal while the FFT approach has problems with the stepped C major scale. The LTP algorithm has the lowest accuracy which is caused by the discrete lag determination. Additional processing like up sampling or interpolation techniques could increase the resolution, but this holds for the other approaches as well. The LTP approach also introduces the longest latency due to the prediction time. Figure 1 shows the pitch tracking results of the recorded guitar signal. The PRAAT algorithm returns the smoothest pitch estimate but also shows an octave error. The FFT algorithm shows some kind of flutter in the estimate but in average returns a robust result. The YIN algorithm returns robust pitch estimates though it shows runaway value. The LTP algorithm benefits from the prediction and returns the most robust result. We also tested the pitch trackers with a recorded vibrato signal, again the high D as used for test signal 2. The behavior of the pitch trackers was comparable to the results of Test 2, none of the approaches showed obvious deficits.

5. CONCLUSION

We described four different pitch tracking algorithms and compared them in terms of accuracy, latency and their robustness when used with recorded guitar signals. The accuracy of all presented algorithms was good for synthetic signals, but with real guitar signals problems like octave jumps and frequency fluctuations occurred. The most robust pitch tracker was the LTP prediction approach though it introduces the longest latency which is a major drawback if we want to use it for real-time applications. At this point the YIN algorithm seems to be the most suitable algorithm for real-time single note guitar tracking, though there is still need to improve the robustness of the approach. Therefore further research has to be done to achieve a accurate and very robust pitch estimation, for instance by combination of different approaches.

6. REFERENCES

[1] Udo Zölzer, Ed., *DAFX: Digital Audio Effects*, John Wiley & Sons, Inc., New York, NY, USA, 2002.

[2] G. Peeters, "Music pitch representation by periodicity measures based on combined temporal and spectral representations," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, May 2006, vol. 5, pp. 53 – 56.

[3] Paul Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proc. of the Institute of Phonetic Sciences*, University of Amsterdam, 1993, vol. 17, pp. 97 – 110.

[4] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Trans. on Acoustics, Speech and Signal Processing,*, vol. 22, no. 5, pp. 353 – 362, oct 1974.

[5] Alain de Cheveigné and Hideki Kawahara, "Yin, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, no. 4, pp. 1917–1930, 2002.