POLYPHONIC PITCH DETECTION BY ITERATIVE ANALYSIS OF THE AUTOCORRELATION FUNCTION

Sebastian Kraft, Udo Zölzer

Department of Signal Processing and Communications
Helmut-Schmidt-University
Hamburg, Germany
sebastian.kraft@hsu-hh.de

ABSTRACT

In this paper, a polyphonic pitch detection approach is presented, which is based on the iterative analysis of the autocorrelation function. The idea of a two-channel front-end with periodicity estimation by using the autocorrelation is inspired by an algorithm from Tolonen and Karjalainen. However, the analysis of the periodicity in the summary autocorrelation function is enhanced with a more advanced iterative peak picking and pruning procedure. The proposed algorithm is compared to other systems in an evaluation with common data sets and yields good results in the range of state of the art systems.

1. INTRODUCTION

Polyphonic and multipitch detection is still an unresolved problem in the field of music analysis. A lot of research has been conducted in this area in the last two or three decades and many quite different approaches were developed and published. While the best of these algorithms generally achieve detection accuracies above 60 % in objective evaluations on identical data sets, none of them ever reached values above 70 % [1]. Regarding the multitude of publications in this field it is difficult to give a complete overview. Therefore, the authors would like to point the interested reader to [1, 2, 3] for an extensive survey of state of the art algorithms and only mention the most important ones that served as a basis for this publication in the following paragraphs.

A subgroup of pitch detection algorithms utilises an auditory model as a front-end to mimic the human hearing system, where the unitary pitch perception model from Meddis and O'Mard [4] is the most prominent one. All these models usually include an input filter bank to imitate the frequency resolution capability of the human cochlea. The individual filter channel outputs are then half-wave rectified and lowpass filtered which corresponds to the mechanical to neural transduction of the inner hair cells. Periodicity information per channel is extracted (e.g. using the autocorrelation) and finally summarised or jointly evaluated over all channels.

The basic idea from Meddis' model was used by Tolonen and Karjalainen in their pitch detection algorithm [5], but they drastically reduced the amount of filters in the auditory filter bank and only chose two channels for a maximally efficient implementation. The redundancy in the resulting overall summary autocorrelation function (SACF) was then removed by simply stretching the SACF by integer factors and subtracting it from itself. The analysis procedure is computationally efficient and straight-forward to implement but the detection accuracy can not compete against recently developed methods.

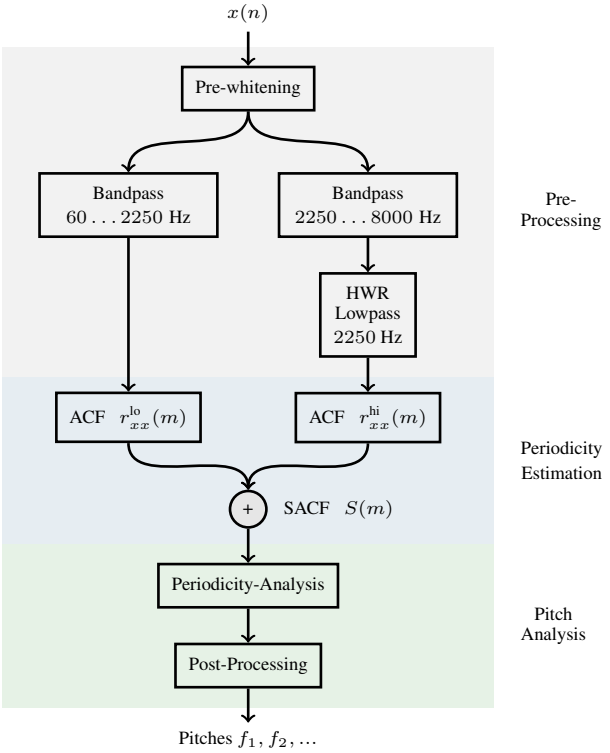


Figure 1: Block diagram of the presented pitch detection algorithm.

When it comes to the detection of multiple pitches with an auditory motivated front-end, one also has to consider the extensive research done by Klapuri [6, 7]. He uses an auditory model to split the input signal into several channels and periodicity information is retrieved from the sum of the individual channel spectra. The subsequent analysis process is looking for peaks with a strong corresponding harmonic series and iteratively removes the strongest series from the spectrum while selecting its base peak as a pitch candidate. The big filter bank (around 70 channels) and complex analysis induce high computational costs but the detection accuracies are good.

In this paper, a two channel auditory front-end like the one from Tolonen is used but the analysis of the periodicity information is replaced by a more advanced iterative peak picking and pruning procedure comparable to the one from Klapuri. Local

maxima in the SACF are detected and periodicity saliencies are calculated by summing the amplitudes at all integer multiples of a peak. High salience values will indicate a strong periodicity and the relating base period of the series can be assumed to be a good pitch candidate. A similar method has already been published by the same authors in [8] but the retrieved pitches were solely used as input for a chord detection and the whole algorithm was never optimised and evaluated in the context of multipitch analysis. Although it still shares the same basic idea, the implementation details and parameters changed a lot while the focus was shifted towards a pure polyphonic pitch detector.

In the following Section 2, the new algorithm will be described in detail followed by an evaluation with three well known data sets in Section 3, including a comparison with the state of the art approach from Benetos [9]. Section 4 will complete the paper with a summary and outlook to future developments.

2. PITCH DETECTION ALGORITHM

The block diagram of the presented pitch detector is depicted in Fig. 1 and in its underlying structure it is identical to the system of Tolonen [5]. Regarding the Pre-Processing and Periodicity Estimation stages, the main modification is a different parametrisation of the auditory front-end. However, the subsequent Pitch-Analysis block has been completely replaced by an iterative method. All signal processing is performed in overlapped blocks $x(n)$ of length N and the hop size between successive blocks N_h is set to $N/4$.

2.1. Pre-processing

The incoming signal block $x(n)$ is first of all processed by a pre-whitening filter. A signal model is estimated by linear prediction and inverse filtering with the model coefficients yields the pre-whitened input block with an equalised spectral envelope. To achieve a higher resolution in low frequency regions, the filter coefficients are determined by warped linear prediction (WLP) [10]. The WLP model was chosen to be of order 8 with a warping coefficient of 0.72 and the loss of signal energy by the filtering operation was compensated by comparing the overall power per block before and after the filter.

Afterwards the signal is split in two bands. The low channel bandpass filtering is realised by the sequential application of a lowpass and highpass at 2250 Hz and 60 Hz, respectively. The high channel bandpass is formed by a highpass at 2250 Hz followed by a lowpass at 8000 Hz. After half-wave rectification of the high channel signal, another lowpass at 2250 Hz is applied. All filters are second order IIR butterworth types [11] and the filtering is done per block in forward and backward directions to compensate for group delay but also to achieve steeper slopes. Finally, an individual periodicity estimation is performed in both channels.

2.2. Periodicity estimation

The autocorrelation function (ACF) is a common way to determine the periodicity of a signal and it has been frequently used to retrieve pitch information in the past. By using the Wiener-Khintchine theorem it can be efficiently calculated in the frequency domain as the inverse Fourier transform of the power spectrum. To avoid cyclic convolution from the DFT and to respect that the length of an autocorrelation sequence is $N_r = 2N - 1$, the input block has to be zero-padded to N_r before applying the DFT. In this case N_r is chosen to be $N_r = 2N$ (nearest power of two for an efficient FFT implementation). The input block $x(n)$ is first weighted by a Tukey (tapered cosine) window with a control parameter $\alpha = 0.4$ and after appending N zeros the resulting vector

$$\mathbf{x_p} = \overset{N_r \times 1}{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}, \qquad N_r = 2N \tag{1}$$

can be used to calculate the autocorrelation

$$\mathbf{r_{xx}} = \text{IDFT}\left(|\text{DFT}(\mathbf{x_p})|^2\right). \tag{2}$$

By replacing the square in (2) with a parameter γ

$$\mathbf{r_{xx}} = \text{IDFT}\left(|\text{DFT}(\mathbf{x_p})|^\gamma\right) \tag{3}$$

the ACF is non-linearly distorted and the amount of distortion can be easily adjusted. In the presented algorithm $\gamma = 0.6$ was used. The ACF is calculated individually in the high and low channel and the summary autocorrelation function (SACF)

$$S(m) = r_{xx}^{\text{lo}}(m) + r_{xx}^{\text{up}}(m), \quad m \in [0, \ldots, N_r], \tag{4}$$

with the time lag index m, is further analysed in the next step to extract the pitch information.

One interesting feature of the ACF in general, and also of the SACF as used in this paper, is the fact that its shape is approximately independent from the spectral envelope of the input signal. In Fig. 2 the SACFs of four harmonic signals with an identical fundamental frequency of 440 Hz but different spectral envelopes are shown. Although some of the signals have quite different partial amplitudes or even missing partials in the spectrum, the main period is clearly visible in all SACF plots and the corresponding peaks have an identical amplitude gradient. This is particularly beneficial for iterative detection approaches. Detected peaks have to be removed before the next iteration starts and the wrong estimation of peak amplitudes in the case of overlapping peaks is a common difficulty for algorithms that perform this kind of processing in the spectrum. In the SACF, the envelope is highly predictable and can be simply determined by fitting a smooth curve through the peak amplitudes.

2.3. Periodicity analysis

The SACF contains all the periodicity information from the input signal emphasised by the various pre-processing steps. The challenge is to analyse the SACF and to transfer this periodicity information to distinctive pitches. In [5] the SACF was iteratively stretched and subtracted from itself to remove redundant information. The remaining peaks above a final threshold eventually mark the most prominent fundamental periods in the signal. While being computationally efficient and easy to implement, the repeated reductions are not very specific as with increasing stretch factors the widening and subtraction of the SACF increasingly deforms the relevant peaks. Therefore, we propose to replace this analysis step with an iterative peak picking and pruning approach.

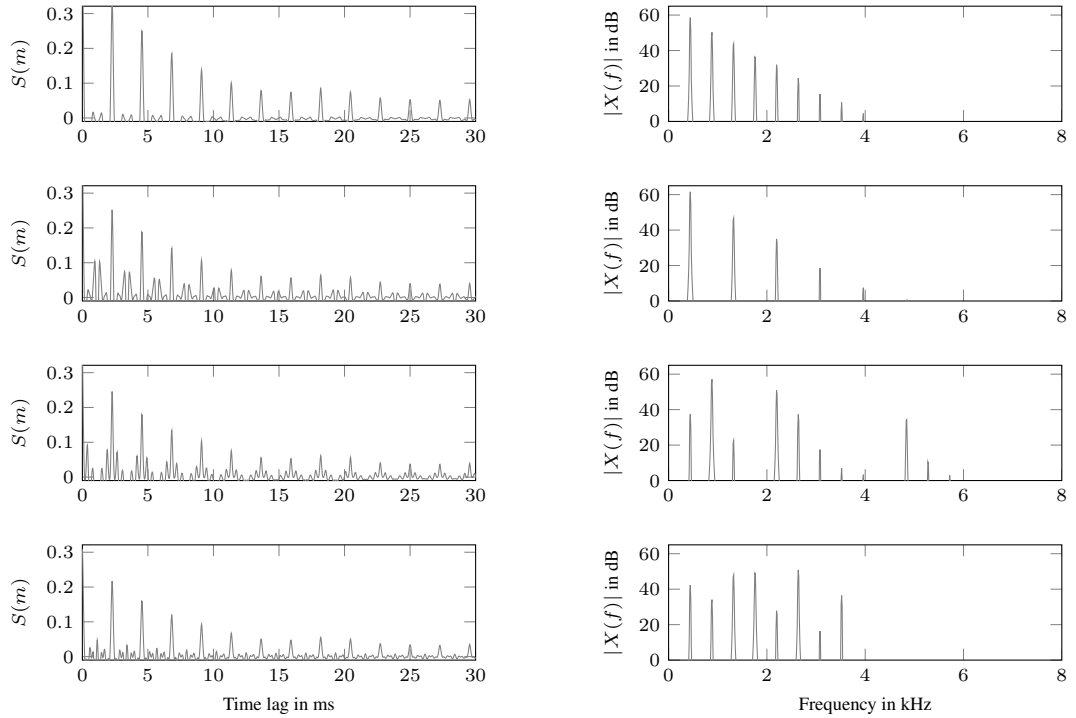


Figure 2: Outputs of the summary autocorrelation function (SACF) for input signals with different spectral envelopes. Fundamental frequency of all signals is 440 Hz which corresponds to a period of 2.27 ms.

2.3.1. Periodicity salience

Initially, a set of all local maxima (or peaks)

$$\mathbb{M} = [m_1, m_2, \ldots, m_i, \ldots, m_M], \quad m_{\text{lo}} < m_i < m_{\text{hi}} \tag{5}$$

above a threshold δ_1 in the SACF is identified, where m_{lo} and m_{hi} are the minimum and maximum lag values to take into account as fundamental frequencies and $i \in [1, \ldots, M]$ is the index of the maximum in the list. For every maximum a corresponding periodicity salience will be calculated by summing the SACF values at all integer multiples. A high salience will indicate that the investigated maximum is the base peak of a strong series in the SACF and hence, is a good candidate for a fundamental period.

The whole process is shown as pseudo code in Algorithm 1 and described in detail in the following paragraphs. The outer loop iterates over all detected maxima in $\mathbb{M}$. A tolerance value $\Delta_m = 4 + {}^{m_i}/_{25}$ is calculated for the maximum m_i and the corresponding salience s_i is initialised with the SACF amplitude $S(m_i)$ of the base peak. The peak counter k_i is set to one and the exact position of the first maximum $\hat{m}_{i,1}$ is initialised with m_i.

The inner loop iterates over all integer multiples k of the base peak, whereas k is bound to the nearest integer $[{}^{m_{\text{max}}}/_{m_i}]$ and m_{max} denotes the maximum lag that is considered being a multiple. The k-th multiple of m_i in the series is estimated to appear at

$$m_{i,k} = \hat{m}_{i,k-1} + m_i, \qquad k = 1, 2, 3, \ldots, [\frac{m_{\text{max}}}{m_i}] \tag{6}$$

and the exact location

$$\hat{m}_{i,k} = \underset{m_{i,k} \pm \Delta_m}{\text{argmax}} \; [S(m)] \tag{7}$$

is retrieved as the local maximum of $S(m)$ in a range of $\pm\, \Delta_m$ around the approximate position. If the periodicity error

$$\Delta_{\hat{m}_{i,k}} = |m_{i,k} - \hat{m}_{i,k}| \tag{8}$$

is smaller than the tolerance Δ_m, a valid peak in the current series is detected. Its amplitude $S(\hat{m}_{i,k})$ is added to the periodicity salience

$$s_i = s_i + S(\hat{m}_{i,k}) \tag{9}$$

and the counter of detected peaks in the current series

$$k_i = k_i + 1 \tag{10}$$

is incremented by one.

After the border m_{max} is reached and $m_{i,k} > m_{\text{max}}$ for the current k, a refined base peak position

$$\hat{m}_i = \frac{1}{k_i} \sum_{k \in \mathbb{K}} \frac{\hat{m}_{i,k}}{k} \tag{11}$$

can be calculated by taking the mean value of all peak positions in the series, where $\mathbb{K}$ is the set of all k where the maxima satisfy Eq. (8). This even allows sub-sample accuracy in the period measurement and therefore, an increased frequency resolution in particular for high frequencies. Otherwise, the precision would be limited by the sample time $T_s = {}^1/_{f_s}$. Furthermore, the saliencies

$$s_i = s_i \cdot \left(\frac{k_i}{\frac{m_{\text{max}}}{\hat{m}_i}} \right)^2 \tag{12}$$

// iterate over all maxima m_i in $\mathbb{M}$
for $i \leftarrow 1$ **to** M **do**
  $\Delta_m \leftarrow 4 + {}^{m_i}/_{25}$
  $s_i \leftarrow S(m_i)$
  $k_i \leftarrow 1$
  $\hat{m}_{i,1} \leftarrow m_i$

  // iterate over all multiples $m_{i,k}$
  for $k \leftarrow 2$ **to** $[{}^{m_{max}}/_{m_i}]$ **do**
    $m_{i,k} \leftarrow \hat{m}_{i,k-1} + m_i$
    $\hat{m}_{i,k} \leftarrow \text{argmax}_{m_{i,k} \pm \Delta_m} [S(m)]$
    $\Delta_{\hat{m}_{i,k}} \leftarrow |m_{i,k} - \hat{m}_{i,k}|$

    // if peak error is smaller than tolerance
    if $\Delta_{\hat{m}_{i,k}} < \Delta_m$ **then**
      $s_i \leftarrow s_i + \text{SACF}(\hat{m}_{i,k})$
      $k_i \leftarrow k_i + 1$
    end
  end
  $\hat{m}_i \leftarrow \frac{1}{k_i} \sum_{k \in \mathbb{K}} \frac{\hat{m}_{i,k}}{k}$
  $s_i \leftarrow s_i \cdot \left(\frac{k_i}{m_{\max}/\hat{m}_i} \right)^2$
end

Algorithm 1: Calculation of periodicity saliencies s_i for a set of detected maxima $\mathbb{M}$.

are weighted by the number of detected peaks over the number of potentially available peaks below $m_{\max}$. This factor can be interpreted as a measure of how complete a series is and it goes down to zero if only a few random or even no multiples were found.

The maximum $m_i^\star$ with the strongest salience s_i is then finally chosen as the first pitch candidate and the corresponding fundamental frequency

$$f_1 = \frac{f_s}{\hat{m}_i^\star} \tag{13}$$

is calculated with the help of the sampling frequency f_s.

2.3.2. Peak pruning

After selecting the strongest maximum, the corresponding peak series (base peak and multiples) has to be removed from the SACF before proceeding to the next iteration. The pruning procedure is shown in pseudo code in Algorithm 2. The detection of multiples in a series is identical to the one in Algorithm 1 and its detailed description is found in the previous section.

In Sec. 2.2 it was already mentioned that the envelope of a peak series is well predictable and in this case it is assumed to follow an exponential curve

$$\hat{S}(m) = a \cdot e^{b \cdot m}, \tag{14}$$

where the parameters a and b are estimated by a curve fitting algorithm. After erasing the base peak, all exact positions of the multiples are identified and removed. The removal of a peak with the removePeak() function in the pseudo code works as follows:

1. Find the inflection points left and right of $m_{i,k}^\star$ to determine the width of the peak.
2. Retrieve the estimated peak amplitude.

$\Delta_m \leftarrow 4 + {}^{m_i^\star}/_{25}$
$\hat{m}_{i,1}^\star \leftarrow m_i^\star$
// remove base peak $m_i^\star$
removePeak($m_i^\star$)

// remove all multiples of $m_i^\star$
for $k \leftarrow 2$ **to** $[{}^{m_{max}}/_{m_i^\star}]$ **do**
  $m_{i,k}^\star \leftarrow \hat{m}_{i,k-1}^\star + m_i^\star$
  $\hat{m}_{i,k}^\star \leftarrow \text{argmax}_{m_{i,k}^\star \pm \Delta_m} [S(m)]$
  $\Delta_{\hat{m}_{i,k}^\star} \leftarrow |m_{i,k}^\star - \hat{m}_{i,k}^\star|$

  // if peak error is smaller than tolerance
  if $\Delta_{\hat{m}_{i,k}^\star} < \Delta_m$ **then**
    removePeak($\hat{m}_{i,k}^\star$)
  end
end

Algorithm 2: Pruning of a periodic series from the SACF $S(m)$ starting with the most salient maximum at $m_i^\star$.

3. Create a tapered cosine window $w(m)$ (Tukey window) which spans the whole width of the peak (parameter $\alpha = 0.2$) and is zero elsewhere.
4. Remove the peak by multiplication with a properly scaled inverse window

$$w(m)' = \left(1 - \frac{\hat{S}(\hat{m}_{i,k}^\star)}{S(\hat{m}_{i,k}^\star)} \cdot w(m) \right) \tag{15}$$

$$S(m) = S(m) \cdot w'(m), \tag{16}$$

where $\hat{S}(\hat{m}_{i,k}^\star)$ is the expected peak amplitude determined by the curve fitting as in (14). In the case that $\hat{S}(\hat{m}_{i,k}^\star) > S(\hat{m}_{i,k}^\star)$, the quotient has to be bound to one to avoid a negative window amplitude.

After the removal of all peaks in the series the next iteration starts and the whole process is repeated until a certain break condition is met.

2.3.3. Break condition

There are two possible conditions to stop the iterations for the current frame and to proceed to the next one. First condition is to limit the average number of iterations to the expected count of simultaneous note events (polyphony). As this is usually unknown and may also change drastically throughout a musical piece, the polyphony alone is not a sufficient criterion. Therefore, iterations will also stop when the strongest salience does not any more excel a threshold δ_2, where usually $\delta_2 > \delta_1$.

2.3.4. Parameters

From the previous algorithmic description it could already be seen that there are a lot of free parameters. Most of them are quite empirical and can only be tweaked manually without any mathematical or physical relationship. This makes it difficult to give an optimal parameter set. However, the parameters in Table 1 turned out to yield good results with all data sets during the development process and also in the later evaluation. All parameters were determined for a sampling frequency of 44.1 kHz.

Description	Param.	Value
Block length	N	4096
Hop size	N_h	1024
Peak position tolerance	Δ_m	$4 + m_i/25$
Peak detection threshold	δ_1	0.025
Salience threshold	δ_2	$0.12 \approx 5\,\delta_1$
Max. number of iterations	-	6
Min. period of base peaks m_i	m_{lo}	30
Max. period of base peaks m_i	m_{hi}	735
Max. period of multiples $m_{i,k}$	m_{max}	2048

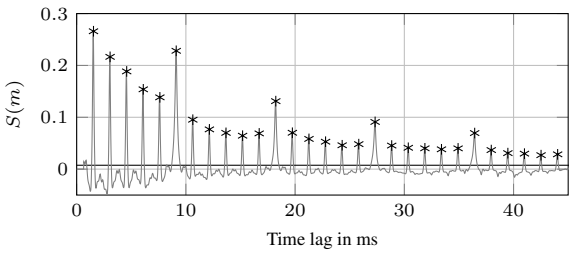
Table 1: Parameters of the periodicity analysis ($f_s = 44.1\,\text{kHz}$).

2.3.5. Example

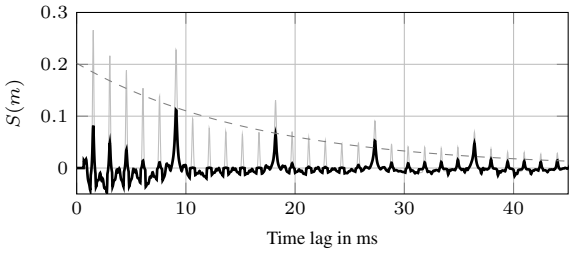
In Fig. 3 the peak picking and pruning procedure is depicted for a single iteration on a sample signal containing two harmonic tones with fundamental frequencies of 110 Hz and 659 Hz. The peaks of the strongest detected series in the first iteration are marked by an asterisk in Fig. 3a). This series is then removed in Fig. 3b) under the assumption of the estimated envelope which is drawn as a grey dashed line. Now, the residual thick black curve mainly contains periods of the lower fundamental frequency and the corresponding strongest series is chosen in Fig. 3c). Due to the smooth and well approximated envelope of the peak amplitudes it is possible to separate these tones even though the two series completely overlap.

2.4. Post-processing

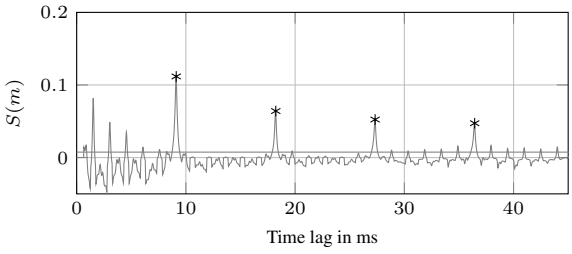
A simple post-processing filter was used to remove isolated and spurious detections with the length of a single frame. It is also intended to fill single frame gaps in otherwise stable detections over various frames. Despite its simplicity it turned out to be very effective. Applied to algorithms with many spurious false positives the post-processing has the ability to drastically raise the Precision with only negligible decrease of the Recall values.



(a) Selected peak series with the strongest salience



(b) After removal of the first series



(c) Selection of the next series

Figure 3: Peak picking and pruning in the SACF of a signal with fundamental frequencies of 110 Hz and 659 Hz. Subplot a) shows the selected peak series with the strongest salience in the first iteration which is then removed in b), where the dashed line shows the estimated envelope. The lower frequency series stays intact after the removal. In the residual c) the next series will be selected.

3. EVALUATION

3.1. Data sets

The pitch detection algorithm, described in the previous chapter, has been evaluated with three different data sets. All of them are established in the community and have been used to evaluate various other algorithms in the past:

- *Bach10 Data Set* [12] consists of ten excerpts from several J.S. Bach chorales played by violin, clarinet, saxophone and bassoon. Matlab data files with fundamental frequencies and onset/offset times are supplied as ground truth.
- *MIREX Multi-F0 Woodwind Development Data Set* [13, 14] is the recording of a woodwind quintet (flute, oboe, clarinet, horn and bassoon) with the respective pitch information as a MIDI file. The whole recording has a length of 9 minutes and is one of the pieces used in the evaluation of the annual *MIREX Multiple Fundamental Frequency Estimation and Tracking* task. Only a 30 second training snippet is publicly available and was used for this evaluation.
- *TRIOS Score-aligned Multitrack Recordings Data Set* [15] is a collection of 4 multitrack recordings of short extracts from classical trio pieces performed by piano, string and several wind instruments. It also includes an additional recording of the famous *Take Five* jazz piece played by piano, saxophone and drums.

Regarding the density and polyphony of the music, the Bach10 data set is the most simple one. Its pieces are played by a quartet of monophonic instruments and therefore, have a maximum polyphony of four. The same holds true for the MIREX piece, but as it is played by a quintet, its polyphony is limited to five. The most complex data set is TRIOS as it contains two monophonic instruments mixed with a difficult piano track which alone induces a high polyphony. All input signals are available at a sample rate of 44.1 kHz and were mixed down to mono if necessary. Additional normalisation to a mean sample power of one was applied to allow an almost data set independent parametrisation of the algorithms.

3.2. Metrics

For the calculation of the evaluation metrics, the amount of true positive, false positive and false negative detections were counted on a frame basis of $10\,\mathrm{ms}$ and accumulated over all songs in a data set. Based on these values the standard metrics Precision, Recall and F-measure were retrieved [13]. If the pitch detector output was given as a set of fundamental frequencies, they were converted and rounded to the closest integer MIDI value.

3.3. Algorithms and parameters

Besides the approach presented in this paper, three other algorithms were investigated. The algorithm from Tolonen [5] shares the same front-end as the presented approach. Hence, its purpose is to show if the new iterative analysis of the SACF yields any advantages. The algorithm from Klapuri [7] is also based on an auditory front-end but uses a far more complex filter bank as input stage. Its iterative analysis procedure is comparable to the presented one. Both algorithms were carefully implemented by the authors of this paper in Matlab.

Finally the publicly available Matlab implementation[1] of a recent algorithm presented by Benetos [9] is included. It is among the best algorithms that have participated in the MIREX campaign in the last years and well suited to compare the presented algorithm to a current state of the art system. Regarding its processing principle it is completely different to the other systems in this evaluation. The algorithm takes the log-frequency spectrogram matrix as input and tries to find a suitable factorisation into an activation matrix and accompanying spectral templates. In a training stage the spectral templates can be initialised with pre-trained spectra to guide the later factorisation process.

The three reference systems were parameterised as recommended in the respective papers. In particular:

- Benetos: sparsity for pitch activation $s_z = 1.05$, sparsity for source contribution $s_u = 1.5$, sparsity for pitch shifting $s_h = 1.1$. Time resolution of the resulting transcription matrix was $40\,\mathrm{ms}$. Final threshold for the transcription matrix was set to $\delta_B = 45$.
- Klapuri: blocklength $N = 4096$, hop size $N_h = 2048$, all other parameters were chosen as proposed in [7].
- Tolonen: blocklength $N = 4096$, hop size $N_h = 1024$, all other parameters as in [5].

All parameters, and primarily the thresholds, were manually tweaked to yield a good balance between Precision and Recall throughout all data sets. Due to the huge amount of parameters it was not possible to iteratively optimize them automatically and it cannot be claimed that they are optimal under all conditions. However, the comparison with previously published evaluations in the next section will validate that the algorithms capabilities are well reflected in our results.

3.4. Results

The detailed results from the evaluation with all data sets are listed in Table 2. Every algorithm was evaluated in 4 different modes. The first block of results is from the pure pitch detector outputs. In the second block, the scores were calculated without taking the absolute octave into account and only the correct detection of the semitones was considered (chroma only). The post-processed results are achieved with the simple post-processing described in Sec. 2.4 and finally the post-processed results are also evaluated with chroma only metrics.

The auditory motivated iterative analysis of Klapuri yields generally better scores than the approach from Tolonen but it does not reach the results from recently developed algorithms. This matches the experience from various other evaluations [7, 16, 17] in the past. However, in absolute values our implementation of Klapuri's algorithm seems to be a few percent worse than reported in the above publications. In contrast the Tolonen algorithm performs a bit better than the implementation from the MIR Toolbox [18] used in [16, 17]. Comparing the post-processed Benetos results in Table 2 with the frame based F-measures in [9] (where a similar post processing was applied), one can see that the values are quite close for the MIREX and TRIOS data set (MIREX: 67.2 %, TRIOS: 66.5 % in [9]). The algorithm has also been evaluated in the context of the MIREX campaign [19] and detailed results are published on the corresponding website [14]. Again, the post-processed results from our evaluation of the Benetos implementation are in the same range. Small deviations of about 5 % may be caused by different parameter settings, thresholds, or in particular different training data. No data set specific training has been conducted during this evaluation and the pre-trained basis spectra from the available Matlab code have been used. However, in [19] it was mentioned that elaborate training with various instruments was performed for the MIREX contribution. After all, one can state that our results of the reference algorithms are plausible and they seem to be properly configured and evaluated.

The presented algorithm with an iterative analysis of the SACF clearly performs much better than the simple stretch and subtract procedure from Tolonen throughout all data sets and metrics. It also yields better results than our implementation of the Klapuri algorithm which uses a similar periodicity analysis but a much more complicated pre-processing. This is a good indication that it is not necessary to rely on a complex auditory model as a front-end. At least it seems possible to drastically reduce the amount of filters for a higher computational efficiency. The proposed system works best on the simple Bach10 data set, where the F-measure is $5.3\,\%$ better than Benetos when post-processing is applied. The results from all algorithms decrease with increasing complexity and polyphony of the music. Finally, on the most complex TRIOS data set, the presented approach and the one from Benetos reach a nearly identical F-measure of 62.9 % and 63.1 %, respectively. On all data sets, the Precision of the presented algorithm is constantly high and only the Recall degrades with increasing polyphony. This indicates a constantly low false positive rate and a slight penalty with highly polyphonic content.

The simple post processing turned out to be very effective and usually increases the Precision by 10-20 % on all algorithms with only minor impact on the Recall values. For future research it might be in particular interesting to see how it compares with more complex post-processing methods like note tracking, e.g. with a hidden Markov model (HMM) as in [20].

To summarize the evaluation, one can say that the presented algorithm with its iterative analysis of the SACF shows a clear advantage over the approach from Tolonen and is more accurate than the algorithm from Klapuri. In fact, the results indicate that the performance is in the range of current state of the art joint estimation approaches like the one from Benetos.

[1] `https://code.soundsoftware.ac.uk/projects/amt_mssiplca_fast`

	standard			chroma only			**post-proc.**			post-proc. + chroma only		
Algorithm	F-meas.	Prec.	Rec.	F-meas.	Prec.	Rec.	**F-meas.**	Prec.	Rec.	F-meas.	Prec.	Rec.
iterSACF	74.0 %	69.3 %	79.3 %	86.8 %	83.5 %	90.4 %	**85.0** %	90.2 %	80.3 %	94.4 %	100.0 %	89.3 %
Benetos[9]	68.4 %	61.6 %	76.8 %	86.4 %	81.7 %	91.7 %	**79.7** %	83.2 %	76.5 %	95.5 %	100.0 %	91.4 %
Klapuri[7]	61.9 %	60.0 %	64.0 %	72.1 %	67.5 %	77.3 %	**68.3** %	73.8 %	63.5 %	86.1 %	100.0 %	75.7 %
Tolonen[5]	61.4 %	61.5 %	61.2 %	72.9 %	70.7 %	75.3 %	**66.8** %	73.6 %	61.2 %	85.5 %	100.0 %	74.7 %

(a) Bach10 data set

	standard			chroma only			**post-proc.**			post-proc. + chroma only		
Algorithm	F-meas.	Prec.	Rec.	F-meas.	Prec.	Rec.	**F-meas.**	Prec.	Rec.	F-meas.	Prec.	Rec.
iterSACF	61.6 %	58.3 %	65.3 %	77.2 %	69.3 %	87.3 %	**73.2** %	83.7 %	64.9 %	90.7 %	100.0 %	83.0 %
Benetos[9]	63.9 %	62.0 %	65.9 %	78.0 %	71.5 %	85.9 %	**69.5** %	76.0 %	64.1 %	91.7 %	100.0 %	84.7 %
Klapuri[7]	51.0 %	50.5 %	51.5 %	68.2 %	60.9 %	77.6 %	**57.0** %	70.7 %	47.7 %	84.7 %	100.0 %	73.5 %
Tolonen[5]	41.4 %	40.5 %	42.3 %	62.9 %	54.2 %	74.9 %	**48.3** %	57.1 %	41.8 %	84.2 %	100.0 %	72.8 %

(b) MIREX data set

	standard			chroma only			**post-proc.**			post-proc. + chroma only		
Algorithm	F-meas.	Prec.	Rec.	F-meas.	Prec.	Rec.	**F-meas.**	Prec.	Rec.	F-meas.	Prec.	Rec.
iterSACF	54.5 %	58.8 %	50.8 %	73.3 %	71.8 %	74.8 %	**62.9** %	82.8 %	50.7 %	83.6 %	100.0 %	71.8 %
Benetos[9]	57.7 %	68.6 %	49.8 %	74.2 %	83.5 %	66.7 %	**63.1** %	86.6 %	49.6 %	79.4 %	100.0 %	65.9 %
Klapuri[7]	45.7 %	52.3 %	40.5 %	60.9 %	59.9 %	61.9 %	**50.5** %	70.7 %	39.2 %	73.6 %	100.0 %	58.2 %
Tolonen[5]	43.0 %	48.0 %	38.8 %	62.4 %	59.7 %	65.3 %	**47.4** %	61.7 %	38.5 %	77.9 %	100.0 %	63.8 %

(c) TRIOS data set

Table 2: Detailed evaluation results grouped by four different evaluation modes: standard rating from the pure pitch detector output, chroma only ratings, ratings with applied post-processing and finally with post-processing and chroma only ratings.

4. CONCLUSION

Starting from the two channel auditory front-end of Tolonen, a new method for the extraction of multiple fundamental frequencies from polyphonic signals was derived. It is based on a novel approach to iteratively extract pitch information from the autocorrelation function. The evaluation proves that the new algorithm is able to yield significantly higher scores than the basic system from Tolonen and also performs better compared to the similar iterative analysis from Klapuri. An average F-measure of 62.9 % was achieved with the TRIOS data set, 73.2 % with the MIREX piece and 85.0 % with the Bach10 data set. These are promising first results in the range of current state of the art algorithms. However, more extensive evaluations are necessary, e.g. in the context of the MIREX campaign, to give an absolute ranking.

One problem of the presented algorithm is its immense amount of parameters that can only be tweaked empirically. Detailed analysis of the parameters, thresholds and their influence on the metrics still has to be done but may be quite time consuming due to the high degree of freedom and existing parameter dependencies. Therefore, it may be interesting to keep the front-end and the advantages of the SACF as described here but apply a joint estimation analysis, as for example non-negative matrix factorisation (NMF) [21] or to make use of probabilistic methods like [9].

5. REFERENCES

[1] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri, "Automatic Music Transcription: Breaking the Glass Ceiling," in *Proc. 13th International Society for Music Information Retrieval Conference*, 2012.

[2] Anssi Klapuri, *Signal Processing Methods for the Automatic Transcription of Music*, Ph.D. thesis, 2004.

[3] Chunghsin Yeh, *Multiple Fundamental Frequency Estimation Of Polyphonic Recordings*, Ph.D. thesis, 2008.

[4] Ray Meddis and Lowel O'Mard, "A unitary model of pitch perception," *Journal of the Acoustical Society of America*, vol. 102, no. 3, pp. 1811–20, Sept. 1997.

[5] Tero Tolonen and Matti Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 708–716, 2000.

[6] Anssi Klapuri, "A perceptually motivated multiple-f0 estimation method," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005.

[7] Anssi Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 255–266, 2008.

[8] Adrian von dem Knesebeck, Sebastian Kraft, and Udo Zölzer, "Realtime System For Backing Vocal Harmonization," in *Proc. of the 14th Int. Conference on Digital Audio Effects*, 2011.

[9] Emmanouil Benetos, Srikanth Cherla, and Tillman Weyde, "An effcient shiftinvariant model for polyphonic music transcription," in *Proc. 6th International Workshop on Machine Learning and Music*, 2013.

[10] Unto K. Laine, Matti Karjalainen, and Toomas Altosaar, "Warped linear prediction (WLP) in speech and audio processing," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1994.

[11] Udo Zölzer, *DAFX: Digital Audio Effects*, John Wiley & Sons, 2nd edition, 2011.

[12] Zhiyao Duan, Bryan Pardo, and Changshui Zhang, "Multiple Fundamental Frequency Estimation by Modeling Spectral Peaks and Non-Peak Regions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2121–2133, Nov. 2010.

[13] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie, "Evaluation of multiple-f0 estimation and tracking systems," in *Proc. of the 10th International Society for Music Information Retrieval Conference*, 2009.

[14] MIREX, "Music Information Retrieval Evaluation eXchange," `http://music-ir.org/mirexwiki/`.

[15] Joachim Fritsch, *High Quality Musical Audio Source Separation*, Master, 2012.

[16] Emmanuel Vincent, Nancy Bertin, and Roland Badeau, "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, Mar. 2010.

[17] Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[18] Olivier Lartillot and Petri Toiviainen, "A matlab toolbox for musical feature extraction from audio," in *Proc. of the 10th Int. Conference on Digital Audio Effects*, 2007.

[19] Emmanouil Benetos and Tillman Weyde, "Multiple-f0 estimation and note tracking for mirex 2013 using an efficient latent variable model," in *Music Information Retrieval Evaluation eXchange (MIREX)*, 2013.

[20] Matti P. Ryynänen and Anssi Klapuri, "Polyphonic music transcription using note event modeling," in *Proc. IEEE-Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005.

[21] Paris Smaragdis and Judith C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003, number 3.