BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 62, No. 4, 2014
DOI: 10.2478/bpasts-2014-0081

Multipitch estimation using judge-based model

K. RYCHLICKI-KICIOR* and B. STASIAK

Institute of Information Technology, Lodz University of Technology, 215 Wolczanska St., 90-924 Łódź, Poland

Abstract. Multipitch estimation, also known as multiple fundamental frequency (F_0) estimation, is an important part of the Music Information Retrieval (MIR) field. Although there have been many different approaches proposed, none of them has ever exceeded the abilities of a trained musician. In this work, an iterative cancellation method is analysed, being applied to three different sound representations – salience spectrum obtained using Constant-Q Transform, cepstrum and enhanced autocorrelation result. Real-life recordings of different musical instruments are used as a database and the parameters of the solution are optimized using a simple yet effective metaheuristic approach – the Luus-Jaakola algorithm. The presented approach results in 85% efficiency on the test database.

Key words: MIR, fundamental frequency estimation, multi F0, multipitch, polyphony.

1. Introduction

Multiple fundamental frequency (F_0) estimation is a low-level task defined within the Music Information Retrieval (MIR) field. It forms a foundation for more complex and high-level problems, such as Audio Chord Estimation, Audio Melody Extraction or Real-time Audio to Score Alignment [1].

This task is much different from recognizing only one fundamental frequency – a simpler task with numerous practical applications, i.a. in pitch tracking for query-by-humming search interface [2] or in speech emotion recognition [3]. More similar, yet distinct task is a melody extraction from polyphonic music signal. Although many different pitches can be detected there, mostly the main pitches – constituting a melody – are taken into consideration [4].

The main goal of the multi F_0 estimation task is to detect correct fundamental frequencies in a signal generated by several independent, concurrent sound sources. The number of the sources can be known (i.e. algorithm always tries to estimate the known number of fundamental frequencies) or not. The latter problem is more complex and involves an additional step called *polyphony inference*. This process is not performed in this work, as the number of the sources is known [5].

Most of the multiple fundamental frequency estimation approaches rely on the spectral analysis. The whole problem could be trivial if the analysed signals were composed of the sums of simple sine waves (i.e. pure tones). This is not the case, however, due to a complicated nature of sound spectra, generated by musical instruments.

Such a spectrum typically consists not only of the fundamental frequency, but also its *partials*, sometimes called *harmonics*. Partials are frequencies that can be calculated using the following formula:

$$f_i = (i+1)f_0, \tag{1}$$

where f_i represents the consecutive partials, i is the i-th partial number and f_0 is the fundamental frequency.

In this work, it is assumed that the first partial is f_0, the second partial is f_1, and so on. Equation (1) describes the idealized case that is often slightly different from the reality [5].

What makes the multi F_0 task difficult is that the fundamental frequency does not always result in the strongest component in the sound spectrum and, more generally, partials do not follow the intuitive rule that the higher the partial is, the weaker magnitude in the spectrum it has. An interesting example is a clarinet – the third partial is often much stronger than the second partial. The example spectrum is shown in Fig. 1.

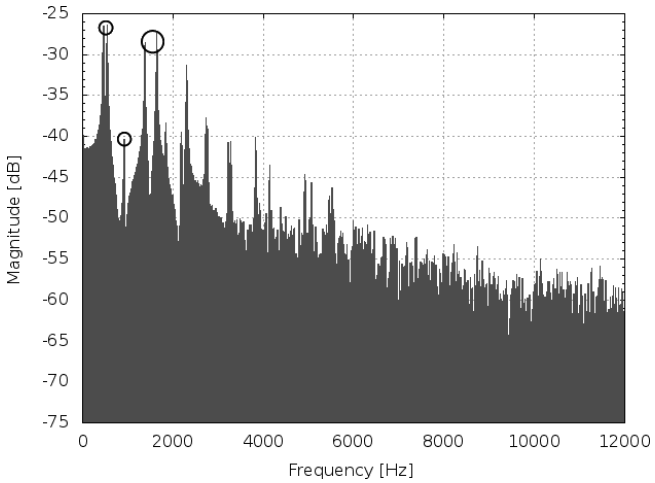

Fig. 1. A spectrum of a sound containing two notes (F#4 and A#4) played on a clarinet. The circles depict consecutive partials of both of them (F_0, F_1 and F_2)

2. Known approaches

The multipitch estimation problem has received many different solution proposals [6]. Multiresolution Fast Fourier Transform (MRFFT) has been used [7] as a compromise between

*e-mail: krzysztof.rychlicki-kicior@dokt.p.lodz.pl

good frequency resolution and good time resolution that results in decreasing the number of overlapping partials. In that approach, pair-wise analysis of spectral peaks is used to find multiple F_0 [7]. Constant-Q Transform (CQT), which is described later, is a very important part of our approach, and it is similar to MRFFT, since both approaches rely on non-linear representation of signal spectrum [1].

The joint estimation approach was applied i.a. by Klapuri [8]. It is described more detailed in the following sections. In Yeh's work [9] much more complex solution was developed, including estimation of the noise level and detecting the number of sound sources. Estimation of noise level removes unnecessary information from the signal, and as a result, decreases a degree of false information about fundamental frequencies and their partials. Polyphony inference (detection of the number of the sound sources) was also analysed, as it is one of the crucial challenges in the general problem of multiple F_0 estimation. Yeh's approach was presented during the MIREX 2007 contest and it achieved accuracy of 65%.

2.1. Different types of sound representation. Before any frequency can be selected, the sound in its basic form, represented in time domain, must be transformed to another domain, since usefulness of time representation for the multiple-pitch estimation is low [5]. Different forms of frequency domain representation are a popular choice – from regular spectrum, obtained with the Discrete Fourier Transform (DFT), up to more specialized forms, such as the MRFFT or cepstrum.

The simplest spectral approach relies on finding the most powerful frequency components. Unfortunately, it does not take partials into account, so if distinct sounds are played with different volume (energy), then besides the fundamental frequency of the first sound, its partials might be selected, whereas the F_0 of the other sound might be omitted. Therefore, such an approach is not used often.

Instead of using the *power* of a frequency component from the spectrum, much more useful descriptor is a *salience*. Salience is a measure that describes the power of a frequency component much better in many MIR-related applications [8]. The difference with the regular power of a frequency component lies within the definition – salience of a given frequency depends also on the power of its partials:

$$s(\tau) = \sum g(\tau, m)\,|Y(f_{\tau,m})|, \tag{2}$$

where Y is the sound spectrum, $f_{\tau,m}$ represents a certain frequency corresponding to the given τ and $g(\tau, m)$ is a weight function that decreases the significance of the further partials. The exact form of the function (2) depends on parameter values, which may be a subject of optimization. M defines the number of partials to be summed and τ represents a *lag*, which is directly related to the frequency component:

$$\tau = \frac{f_s}{f}, \tag{3}$$

where f_s represents a sample rate of the input signal and f is a given frequency.

Salience is a much better representation of the power of the frequency, because it is a weighted sum of powers for all partials of the given frequency. Despite yielding better outcomes than the simple power-based approach, unfortunately it still can give inappropriate results. Often, the second or the third partials are returned, if one of the sounds is louder than the other [8].

The salience approach has been widely used, e.g. by Klapuri [8]. However, in this work, two additional sound representations are also used, in order to increase efficiency: cepstrum and enhanced autocorrelation.

Cepstrum is a transform of a signal that has received much recognition, especially in the analysis of the human speech. It is usually associated with spectrum of the signal and defined using the following formula:

$$C(n) = \left| DFT^{-1} \left\{ \log \left(\left| DFT\left\{x(n)\right\}\right|^2 \right) \right\} \right|^2, \tag{4}$$

where $x(n)$ is the n-th sample of the signal and the DFT is the Discrete Fourier Transform. However, it should be noted that cepstrum may be used more generally, with different transforms.

Within the Music Information Retrieval field, cepstrum is used mostly to recognize the single fundamental frequency of the signal. It does not mean, however, that more complicated analysis process cannot utilize this representation.

Another sound representation is a result of an enhanced autocorrelation. Autocorrelation, like cepstrum, is used to find a single fundamental frequency in a signal. Basic autocorrelation is correlation of a discrete signal with itself:

$$R(\tau) = \lim_{N\to\infty} \sum_{n=0}^{N-1} x(n)x(n-\tau), \tag{5}$$

where τ denotes the lag (in seconds), and x_n denotes the n-th sample of the signal.

The first maximum of the autocorrelation function is then used to calculate the fundamental frequency of the signal:

$$F_0 = \frac{f_s}{\tau_{\max}}, \tag{6}$$

where f_s is the sampling rate and the $\tau_{\max}$ is the first maximum of the lag function.

Enhanced autocorrelation (EAC) is a modified classic autocorrelation, introduced by Tolonen and Karjalainen [10], as described by Mazzoni [11]. The EAC differs in a few details from the original autocorrelation, e.g. the cube root of the spectral components is computed instead of the square root in the original method. Also, the peak pruning process is applied.

Regardless of the method used for detecting the possible frequency candidates, appropriate methods must be applied, in order to select the correct ones, using the given data sources – salience spectrum, cepstrum or the result of EAC. Regular peak picking (selecting the n strongest components in the data source) usually gives poor results, due to a possibility of choosing partials of one sound over the other one or choosing the incorrect partial of correct sound as a fundamental frequency.

Due to that fact, we have introduced other approaches. They have been applied in order to resolve the problem with too strong partials. Both methods described below were initially applied only to salience.

Iterative cancellation has been initially proposed as a salience-based method [5]. After finding the strongest component, it is removed from the spectrum, along with the components representing its partials. Therefore, other sounds can be recognized properly, even if they are not as loud as the previously found sounds. This approach gives better results than the regular spectrum peak picking and it is also very fast, due to the quick algorithm of finding the best salience candidates in the spectrum [8]. However, the *overlapping* of the partials is one of the biggest problems in this approach. Overlapping occurs when two sounds have a common partial in the spectrum. This is especially a problem when the frequency resolution of the spectrum is too low and two slightly different partials are placed in the same frequency bin. When the frequency bin is removed for the stronger of two (or more) sounds, other sounds will not have a possibility to use this bin for their salience [8].

This problem is resolved by using the *joint estimation* approach. This method consists of two basic steps. Firstly, a certain number of strongest salience candidates are selected from the spectrum. Then, every possible combination of candidates is cancelled from the spectrum jointly, in order to obtain the smallest residue.

This method does not rely on the order of detection, however it is more computationally expensive, due to the number of combinations to check: binomial coefficient of n and k, where n is a number of preselected salience candidates and k represents the number of sound sources.

3. Proposed approach

The approach applied in this work consists of a few steps. First, the input signal (a sound file) is divided into frames, using the Hanning function for windowing. Next, each frame is analysed, in order to estimate the best possible frequency candidates. The process of frequency candidates selection involves calculating the three sound representations described before: salience spectrum [5], cepstrum and the EAC result.

Although the application of three different sound representations is innovative itself, we have decided to modify also the classic salience spectrum. This kind of spectrum usually employs the standard DFT. However, in this work, the CQT has been applied.

The CQT differs from the regular DFT, in that it results in the spectrum in the logarithmic scale, i.e. frequency bins, which are distributed linearly within the DFT, become distributed logarithmically within the CQT. The frequency of the k-th CQT frequency bin is defined as:

$$F_k = F_{\min} 2^{k/n}, \tag{7}$$

where n is the size of the CQT transform and $F_{\min}$ is the frequency of the first bin in the CQT spectrum.

The importance of the CQT transform stems from the fact that, when compared to the DFT, it gives much more information about the lower band of the analysed frequency range. This is associated with the bins in the lower band being distributed much more tightly than in the upper band. Better low-frequency resolution gives a possibility to detect spectral peaks more precisely, what finally results in better results of the multipitch estimation.

After obtaining all three sound representations, they are transformed using the iterative cancellation method, in order to select the best frequency candidates for each data source. Finally, the additional algorithm, called the *judge*, calculates the final frequencies using all frequency candidates obtained earlier. The general model of the proposed approach is depicted in Fig. 2.

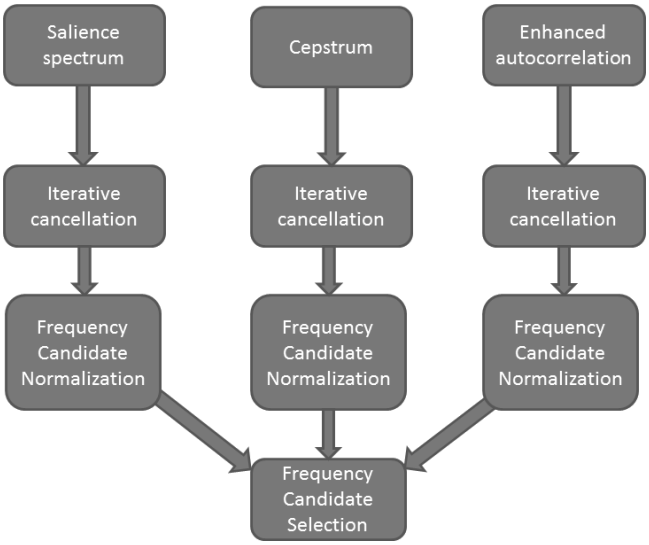

Fig. 2. The proposed approach model

In this work, the iterative cancellation approach has been applied not only to the salience, but also to the cepstrum and the EAC result. The modifications that had to be implemented in the iterative cancellation method, in order to work with two additional methods, are discussed further in this section. Some other changes have been applied, in order to deal with imperfections of Eq. (1) and the overlapping of partials.

The cepstrum and the EAC are also analysed using the iterative cancellation approach. In these cases, however, the iterative cancellation method has been modified. This stems from the meaning of both sound representations and different scales (when compared to regular salience spectrum).

Both cepstrum and the EAC represent functions in lag domain, contrary to frequency-domain spectrum. Since lag is inversely proportional to the frequency:

$$f = \frac{f_s}{\tau} \tag{8}$$

the cepstrum and the EAC plots show higher frequencies closer to zero (in lag scale). Since the few first values of the EAC are very high, we discard them.

As a result, the first few bins in lag-domain plots usually have very high values. Because of that, these bins must be appropriately decreased, adequately to their position in the cepstrum or the EAC.

The very essence of the iterative method is also changed. Whereas in the spectrum case, the strongest frequency component is found and its partials (multiplies) are used for calculating power and for removal, in the cepstrum and the EAC the given sound representation is preprocessed, and then the first nonzero lag component is selected. Then, all multiplies of the selected lag are removed, i.e. all lag bins that belong to the following set:

$$T(\tau) = \{n\tau + \delta : n \in N, \delta \in \{1, 2, ..., WIDTH\}\}, \quad (9)$$

where *WIDTH* is a parameter that is optimized using the Luus-Jaakola algorithm (cf. Subsec. 3.2). The process is performed until there is no data in the cepstrum or the EAC or the assumed number of sources is achieved.

The preprocessing phase of the modified iterative method has a crucial meaning. Since in the EAC and the cepstrum the first found local maximum is selected as a frequency candidate, removing the initial part of the cepstrum or the EAC is very important – otherwise, incorrect candidates may be selected.

Therefore, a special filter function has been constructed. The value of the cepstrum or the EAC bin is nullified when it is smaller than a threshold function value for a given bin. The rule of thumb is that only the strongest bins should be left untouched and the first bins should be treated with a higher rigor. The threshold function is given as follows:

$$Thr(k) = \begin{cases} ak + b : k < BGN \\ c : k >= BGN \end{cases} \quad (10)$$

The *BGN* is the number of the first few bins that are usually higher and the applied threshold must be larger (it is optimized using the Luus-Jaakola approach [12]). The coefficients a, b, and c are defined below. The X means the analysed cepstrum or the EAC result:

$$a = \frac{(\sigma_{coeff}(X) - 1) \cdot \max(X)}{A}, \quad (11)$$

$$b = \max(X), \quad (12)$$

$$c = \sigma_{coeff}(X) \cdot \max(X). \quad (13)$$

The A is the Luus-Jaakola-optimized parameter and σ_{coeff} is given as follows:

$$\sigma_{coeff}(X) = SD \cdot \left(1 - \frac{\sigma(X)}{\max(X)}\right), \quad (14)$$

where *SD* is another optimized parameter (between 0 and 1) and $\sigma(X)$ is the standard deviation of X.

The preprocessing is applied to the original cepstrum or EAC. It uses the simple statistical functions to remove components that are less than a certain percentage of maximal component in the lag spectrum. Moreover, a certain number of the first few components are always removed. An example of a frame before and after processing is shown in Fig. 3. Filtered lag spectrum is transformed using the modified iterative approach.

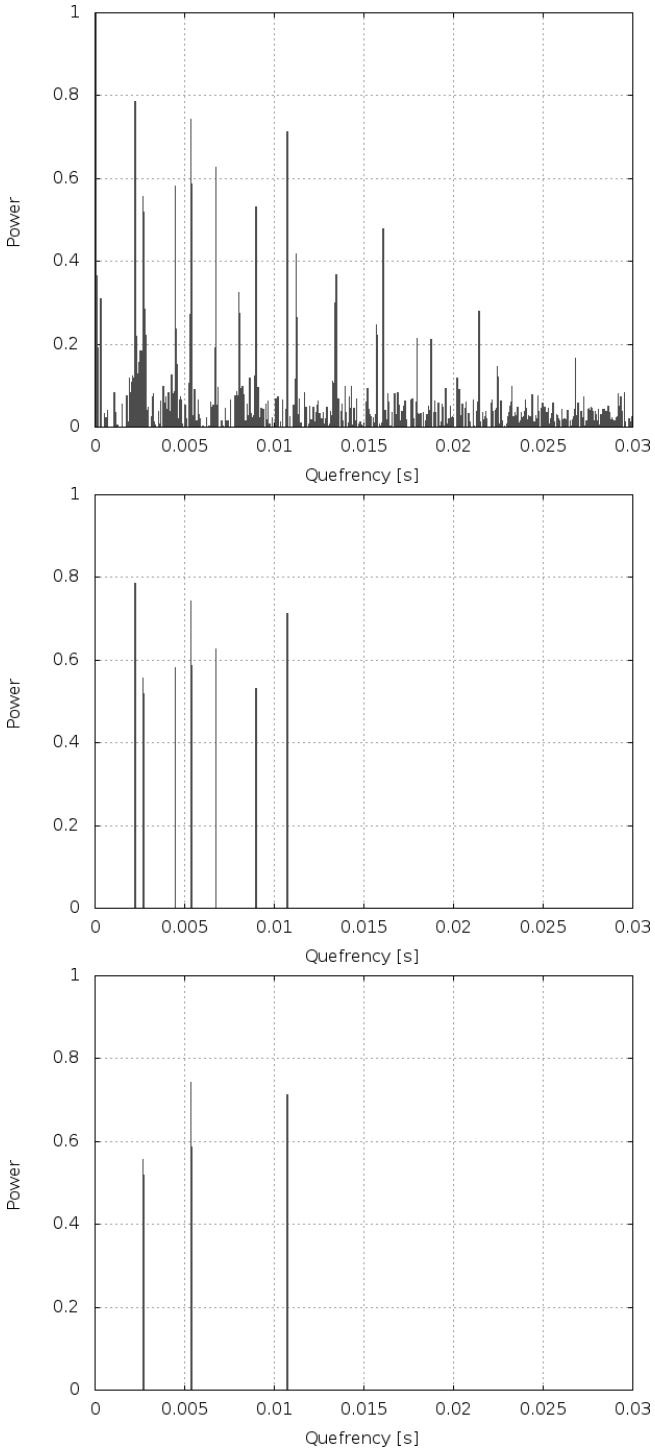

Fig. 3. A cepstrum of an example interval (Alto Sax; F#$_4$, A$_4$). The first cepstrum is the original one. The second depicts the effect of the preprocessing and the third depicts what is left after finding and removing the first frequency component (A$_4$)

The database used to verify the proposed approach has been constructed from several instrument samples (cf. Table 1) from the University of Iowa Musical Instrument Sam-

ples dataset [13]. Basically, the individual sound files (obtained after preliminary cutting procedure yielding a single note within each file) have been mixed to form intervals from 1 to 24 semitones within the range from C4 to F#6 (MIDI note numbers: 60–90). For each file an implementation of Boersma's F_0 estimation algorithm has been applied [14], resulting in a sequence of estimated F_0 values for consecutive time frames. From this sequence the median has been taken as a representation of the true F_0 of the whole file. From all possible combinations of two sound files only the in-tune intervals have been selected.

Table 1
The best F_0 estimation results per instrument

Instrument	Precision [%]
Alto Sax	95.56
Cello Arco	90.79
Clarinet B♭	95.56
Clarinet E♭	93.83
Flute	89.47
Oboe	91.45
Piano	75.58
Viola Arco	97.50
Violin	87.71
Alto Sax & Clarinet E♭	91.67
Alto Sax & Flute	94.94
Clarinet E♭& Flute	94.07
Violin & Flute	92.57
Average	91.59

3.1. Combination of the frequency candidates Since the proposed approach employs several distinct multiple F_0 estimation methods – each of them yielding its own set of frequency candidates – a way of constructing the final set of candidates, based on all these fragmentary sets, must be defined.

Such a method – called hereinafter the *judge* – is a function that takes a vector of lists of frequency candidates and returns the one, final list of frequencies. Each list contains a few frequency candidates. Each candidate is described by a frequency (in Hz) and a power. The meaning of a candidate's power depends on the method. Due to the differences in meaning of power (and the typical value ranges), the power normalization process of the frequency candidates is used, in order to be able to compare the power of the candidates. Since all the samples are considered to have two sounds (pitches) and three data sources are used, in this work the frequency candidate sets analysed by the judge have six elements (unless methods yield less than two candidates, which is also possible).

Therefore, the whole process of creating the one, final set of frequencies that becomes the result of the multipitch frequency estimation applied to a single window of the signal, may be divided into a few steps:

1) Power normalization (preprocessing) – the maximum of all results from each data source is found and then, all results from a given data source are separately normalized using the following formula:

$$X_{norm} = \frac{X}{\max(X)}, \tag{15}$$

where X is the sound representation (the salience spectrum, the cepstrum or the EAC result).

2) Grouping the frequency candidates – all frequency candidates, having the normalized power, are grouped by frequency, provided that their frequencies are similar. This similarity is understood as follows: the set F contains only the frequencies that are *close* to one another if the following formula is true:

$$\underset{f_A, f_B \in F}{\forall} \left| 1 - \frac{f_A}{f_B} \right| \leq CLOSE, \tag{16}$$

where *CLOSE* is another Luus-Jaakola-optimized parameter.

When the grouping is performed, the new frequency candidate is established, having the average frequency and power of all grouped candidates. The count of the grouped frequencies is also noted – all candidates who have not participated in grouping have the default count of 1.

3) Finally, sorting of all candidates is performed, using a special measure that includes both count and power of the whole set of candidates:

$$f_A < f_B \Leftrightarrow c(f_A) + P \cdot p(f_A) < c(f_B) + P \cdot p(f_B), \tag{17}$$

where c is the count of a given frequency candidate and p is its power. P is the Luus-Jaakola-optimized parameter. Then, n best candidates are chosen as the final result (in this work $n = 2$).

3.2. Optimization method Parameters with the most influence on the algorithm's results have been selected and optimized using the metaheuristic Luus-Jaakola approach [12]. This algorithm uses the stochastic optimization to improve the precision achieved by the proposed method. Classic optimization methods (such as Newton's method) could not be used, because the optimized function, that takes a vector of parameters and returns the global precision ($F : R^N \rightarrow R$) is not guaranteed to be convex nor continuous.

In the Luus-Jaakola algorithm, all parameters are optimized at the same time. It employs simple stochastic optimization by sampling random vectors from uniform distribution. The crucial advantage of this method is the very low number of optimized function calls required for algorithm to work properly, because only one call is required per iteration. This is very important, since one iteration results in performing calculations for the whole database. All parameters are optimized at the same time.

4. Results

The results for all the investigated instruments, i.e. total precision per instrument and the average precision, are depicted in Table 1, while the optimal parameter values are depicted in Table 2 (together with the ranges of optimized parameter

values). Since there are always two notes in the given data sample and two notes are detected by the algorithm, the precision is always equals to the accuracy.

Table 2
Optimal values of the algorithm's parameters

Parameter	Minimum	Maximum	Optimal
WIDTH	2	5	3
BGN	15	50	42
SD	0	1	0.5
M	2	6	4
P	1	4	2.6
CLOSE	0.27	0.33	0.30
A	10	90	52

The results are much better for aerophones (that produce sound using a vibrating column of air) than bowed chordophones (that produce sound using a string made vibrating by a bow), because bowed chordophones often produce sounds where the higher partials have greater power than the fundamental frequency.

The relationship between the interval and error rate is also very clear. The most erroneous intervals are 5, 7 and 12, i.e. a fourth, a fifth and an octave. All these intervals form the basis of the harmonic relationships between sounds and are widely known for their consonance sound. This is a result of sharing multiple partials which is a direct cause of relatively high error levels.

Table 3 depicts distribution of the results to particular methods and their combinations. It presents which method (or combinations of methods) contributes most to the global precision. Despite the crucial differences of constructions of all three sound representations, most of the samples are detected by all of them (over 50%). The CQT salience spectrum is the most efficient method – it has the largest accuracy from the methods alone and gives better results when used in combinations. However, it must be noted that the other methods – the cepstrum, the enhanced autocorrelation and both methods together sum up to over eight percent. The tests have shown that the results strongly depend on the instrument being analysed – sometimes (e.g. clarinet or saxophone) the salience spectrum alone is sufficient, but in other cases (e.g. oboe) different methods vastly improve overall results.

Table 3
Precision divided into particular methods and their combinations

Method	Result
CQT salience spectrum (CSS)	10.67
Cepstrum	0.97
Enhanced Autocorrelation (EAC)	0.18
CSS + EAC	10.31
Cepstrum + EAC	7.11
Cepstrum + CSS	10.56
Cepstrum + EAC + CSS	50.90

Table 4 shows the accuracy of each method for each instrument. Although it is clear that the CQT salience spectrum is the best method, the main goal of using different methods is to improve overall quality of results. For example, in Alto Sax and Cello Arco, two other methods vastly improved the final accuracy. It must be noted, though, that including multiple methods, instead of relying on only one, can have its disadvantages. The main problem is the possibility of excluding the good frequency candidate (by the judge) in favour of incorrect yet "popular" candidates chosen by other methods.

Table 4
The precision of the particular instruments per method

	CQT salience spectrum	Cepstrum	Enhanced autocorrelation
Alto Sax	83.33	80.00	83.33
Cello Arco	67.98	84.21	83.77
Clarinet B♭	93.88	76.67	63.88
Clarinet E♭	90.74	77.16	66.67
Flute	84.21	65.79	63.15
Oboe	84.21	68.75	65.12
Piano	71.51	51.16	50.58
Viola Arco	92.50	81.25	77.50
Violin	77.33	71.61	73.94
Alto Sax & Clarinet E♭	91.67	50.00	58.33
Alto Sax & Flute	94.93	63.92	65.82
Clarinet E♭& Flute	92.22	66.67	70.37
Violin & Flute	80.47	71.48	74.21
Average	**84.99**	**69.90**	**68.97**

The results of both the modified and the original approaches have been compared. The original method [5] for the same dataset achieved the precision of 73%, whereas the proposed method gives the precision of over 91%.

5. Conclusions

In this work, the problem of multiple fundamental frequency estimation has been considered. A modified iterative approach has been applied to the three different sound representations – the salience spectrum, the cepstrum and the enhanced autocorrelation result – and it improved overall precision of the main algorithm.

In the future work a better method of selection of the appropriate frequency candidate (the judge algorithm) must be found, since the precision of the presented approach when the ground truth frequencies where compared to the full frequencies candidate sets (without the judge phase), exceeded 95%. Application of machine learning mechanisms, particularly of different types of classifiers, will be considered, in order to resolve the correct frequency candidate problem. Our approach is also planned to be validated on the basis of a database containing more complicated polyphony.

REFERENCES

[1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchoff, and A. Klapuri, "Automatic music transcription: challenges and future directions", *J. Intelligent Information Systems* 41 (3), 407–434 (2013).

[2] B. Stasiak, "Follow that tune – dynamic time warping refinement for query by humming", *Proc. Joint Conf. New Trends in Audio and Video Signal Processing: Algorithms, Architectures, Arrangements, and Applications* 1, 109–114 (2012).

[3] B. Stasiak and K. Rychlicki-Kicior, "Fundamental frequency extraction in speech emotion recognition", In: *Multimedia Communications, Services and Security, Communications in Computer and Information Science*, pp. 287, 292–303, Springer-Verlag, Berlin, 2012.

[4] J. Salomon, E. Gomez, D.P.W. Ellis, and G. Richard, "Melody extraction from polyphonic music signals", *IEEE Signal Processing Magazine* 31 (2), 118–134 (2014).

[5] M. Davy and A. Klapuri, *Signal Processing Methods for Music Transcription*, Springer-Verlag, Berlin, 2006.

[6] F. Argenti, P. Nesi, and G. Pantaleo, "Automatic music transcription: from monophonic to polyphonic", *Musical Robots and Interactive Multimodal Systems*, pp. 27–46, Springer-Verlag, Berlin, 2011.

[7] K. Dressler, "Multiple fundamental frequency extraction for mirex 2012", 13^{th} *Int. Conf. on Music Information Retrieval* 1, CD-ROM (2012).

[8] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes", *Proc. 7th Int. Conf. on Music Information Retrieval* 1, 216–221 (2006).

[9] C. Yeh, "Multiple fundamental frequency estimation of polyphonic recordings", *Ph.D. Thesis*, Universite de Paris, Paris, 2008.

[10] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model. Speech and audio processing", *IEEE Trans. on Speech and Audio Processing* 8 (6), 708–716 (2000).

[11] D. Mazzoni and R.B. Dannenberg, "Melody matching directly from audio", *2nd Annual Int. Symp. on Music Information Retrieval* 1, 17–18 (2001).

[12] R. Luus and T. Jaakola, "Optimization by direct search and systematic reduction of the size of search region", *American Institute of Chemical Engineers J. (AIChE)* 19, 760–766 (1973).

[13] University of Iowa, "Musical instrument samples dataset", http://theremin.music.uiowa.edu/, access date: 20/01/2013.

[14] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound", *IFA Proceedings* 17, 97–110 (1993).