Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes

Anssi Klapuri
Institute of Signal Processing, Tampere University of Technology
Korkeakoulunkatu 1, 33720 Tampere, Finland
anssi.klapuri@tut.fi

Abstract

This paper proposes a conceptually simple and computationally efficient fundamental frequency (F0) estimator for polyphonic music signals. The studied class of estimators calculate the salience, or strength, of a F0 candidate as a weighted sum of the amplitudes of its harmonic partials. A mapping from the Fourier spectrum to a "F0 salience spectrum" is found by optimization using generated training material. Based on the resulting function, three different estimators are proposed: a "direct" method, an iterative estimation and cancellation method, and a method that estimates multiple F0s jointly. The latter two performed as well as a considerably more complex reference method. The number of concurrent sounds is estimated along with their F0s.

Keywords: F0 estimation, pitch, music transcription.

1. Introduction

Pitch information is an essential part of almost all Western music, but extracting the pitch content automatically from recorded audio signals is difficult. Whereas the spectrogram of a music signal can be calculated straightforwardly using the short-time Fourier transform, computing a "piano roll"-representation which shows polyphonic pitch content as a function of time is non-trivial, and systems trying to do this tend to be very complex.

F0 estimation in polyphonic music has been addressed by many authors. Kashino extracted sinusoid tracks from a music signal and grouped them to sound sources based on acoustic features and musical information [1]. De Cheveigné [2], Tolonen and Karjalainen [3], and Klapuri [5] proposed methods based on modeling the human auditory system. Goto [6] and Davy and Godsill [7] employed a parametric signal model and statistical methods. Smaragdis [8] and Abdallah [9], proposed unsupervised learning techniques to resolve sound mixtures, and Poliner and Ellis [10] introduced a classification approach to the problem.

Here we study a certain type of F0 estimators, where an input signal is first spectrally flattened ("whitened") in order to suppress timbral information, and then the *salience*, or strength, of a F0 candidate is calculated as a weighted sum of the amplitudes of its harmonic partials. More exactly, the salience $s(\tau)$ of a period candidate τ is calculated as

$$s(\tau) = \sum_{m=1}^{M} g(\tau, m)|Y(f_{\tau,m})|, \qquad (1)$$

where $f_{\tau,m} = mf_{\mathrm{s}}/\tau$ is the frequency of the m:th harmonic partial of a F0 candidate f_{s}/τ, f_{s} is the sampling rate, and function $g(\tau, m)$ defines the weight of partial m of period τ in the sum. $Y(f)$ is the short-time Fourier transform of the whitened time-domain signal. [1] The spectral whitening is a straightforward pre-processing operation explained in Sect. 2.1. For convenience, we write $s(\tau)$ as a function of the fundamental period τ instead of the F0 ($= f_{\mathrm{s}}/\tau$).

The basic idea of (1) is intuitively appealing since pitch perception is closely related to the time-domain periodicity of sounds, and the Fourier theorem states that a periodic signal can be represented with spectral components at integer multiples of the inverse of the period. Indeed, formulas and principles resembling (1) have been used for F0 estimation by a number of authors, under different names and in different variants. Already in 1960s and 70s, Schroeder introduced the *frequency histogram* and Noll the *harmonic sum spectrum* (see [11, p.414]). De Cheveigné [2] discusses *harmonic selection* methods and, more recently, Walmsley [12] uses the name *harmonic transform* for a similar technique.

The question of an optimal mapping of the Fourier spectrum to a "F0 salience spectrum" is closely related to these methods. Here, the function $g(\tau, m)$ is learned by a brute-force optimization using a large amount of training material. Based on this, a parametric form is proposed for $g(\tau, m)$.

In this paper, three different methods based on (1) are proposed. The first and simplest is a "direct" method based on evaluating $s(\tau)$ for a range of values of τ and picking the desired number of highest local maxima in it. The second method represents an iterative estimation and cancellation approach, where the maximum of $s(\tau)$ is used to estimate one F0 which is then cancelled from the mixture before estimating the next one. The third method estimates all F0s jointly. For the latter two methods, a technique is proposed for estimating the number of sounds in the mixture. The iterative method admits a very fast implementation which

[1] Defining $s(\tau)$ in terms of the power spectrum instead of the magnitude spectrum would have certain analytical advantages, but this led to clearly inferior F0 estimation results despite of extensive investigation.

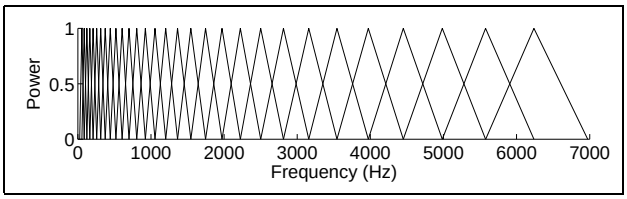

Figure 1. Responses $H_b(k)$ applied in spectral whitening.

does not require evaluating $s(\tau)$ for all period candidates τ. The three methods are evaluated using mixtures of musical instrument sounds, and the results are compared with three reference methods [3], [4] and [5].

2. Proposed methods

This section describes the proposed methods in detail.

2.1. Spectral whitening

One of the big challenges in F0 estimation is to make systems robust for different sound sources. A way to achieve this is to try to suppress timbral information prior to the actual F0 estimation. This can be done by estimating the rough spectral energy distribution (which largely defines the timbre of a sound) and then flattening it entirely or partly by inverse filtering. This process is called spectral whitening and there are several ways of doing it (see e.g. [3]). Here a frequency-domain technique is employed which is easy to implement and leads to good results in practice.

First, the discrete Fourier transform $X(k)$ of the input signal $x(n)$ is calculated in an analysis frame that is Hanning-windowed and zero-padded to twice its length before the transform. Then a bandpass filterbank is simulated in the frequency domain. Center frequencies c_b [Hz] of the subbands are distributed uniformly on the critical-band scale, $c_b = 229 \times (10^{(b+1)/21.4} - 1)$, and each subband $b = 1, \ldots, 30$ has a triangular power response $H_b(k)$ that extends from c_{b-1} to c_{b+1} and is zero elsewhere (see Fig. 1).

Standard deviations σ_b within the subbands b are calculated by applying the responses $H_b(k)$ in the frequency domain:

$$\sigma_b = \left(\frac{1}{K} \sum_k H_b(k) |X(k)|^2 \right)^{1/2}, \qquad (2)$$

where K is the length of the Fourier transform. Next, bandwise compression coefficients $\gamma_b = \sigma_b^{\nu-1}$ are calculated, where $\nu = 0.33$ is a parameter determining the amount of spectral whitening applied. The coeffients γ_b are linearly interpolated between the center frequencies c_b to obtain compression coefficients $\gamma(k)$ for all frequency bins k.

Finally, a whitened spectrum $Y(k)$ is obtained by weighting the spectrum of the input signal by the compression coefficents, $Y(k) = \gamma(k)X(k)$.

2.2. Calculation of the salience function in practice

Calculation of $s(\tau)$ using (1) directly requires evaluating $Y(f)$ for arbitrary frequencies f which is computationally inefficient. Use of the fast Fourier transform becomes possible by replacing $Y(f)$ in (1) by its discrete version $Y(k)$ and by approximating $s(\tau)$ by

$$\hat{s}(\tau) = \sum_{m=1}^{M} g(\tau, m) \max_{k \in \kappa_{\tau,m}} |Y(k)|, \qquad (3)$$

where the set $\kappa_{\tau,m}$ defines a range of frequency bins in the vicinity of the m:th overtone partial of the F0 candidate f_s/τ. More exactly,

$$\kappa_{\tau,m} = [\langle mK/(\tau + \Delta\tau/2) \rangle, \ldots, \langle mK/(\tau - \Delta\tau/2) \rangle], \quad (4)$$

where $\langle \cdot \rangle$ denotes rounding to the nearest integer. It is clear that $\hat{s}(\tau) \approx s(\tau)$ when $\Delta\tau \to 0$. In practice, however, it is useful to set $\Delta\tau$ according to the spacing between successive period candidates τ in order to ensure that all spectral components k belong to the range $\kappa_{\tau,m}$ of at least one period candidate τ when m is fixed. Here we use the value $\Delta\tau = 0.5$, that is, the spacing between fundamental period candidates τ is half the sampling interval. [2]

2.3. Optimization of the weight function

A remaining task is to optimize the function $g(\tau, m)$ so as to minimize the F0 estimation error rate of the system. For this purpose, we generated training material consisting of random mixtures of musical instrument sounds with their reference F0 data. The database from which the samples were drawn is described in more detail in Sect. 3. The mixtures were generated by first allotting an instrument and then a random sound from its playing range, limiting F0s between 40 Hz and 2100 Hz. This two-stage randomizing was repeated until the desired number of sounds was obtained, and the sounds were then mixed with equal mean-square levels. One thousand mixtures of one, two, four, and six sounds were generated, totalling 4000 training instances.

F0 estimation was performed simply by picking P highest local maxima in the function $\hat{s}(\tau)$. The number of F0s in each mixture, P, was given to the estimator along with a 93 ms analysis frame. *Multiple-F0 estimation* error rate is defined as the proportion of reference F0s that were not correctly found. In *predominant-F0 estimation*, the task is to find only one F0 in each mixture. In this case, the maximum of $\hat{s}(\tau)$ was taken and judged correct if it matched any of the reference F0s in the mixture. A correct F0 estimate was defined to deviate less than 3% from the reference. The criterion to be minimized in the optimization was the average of multiple-F0 and predominant-F0 estimation error rates in different polyphonies.

Two different factorized forms of $g(\tau, m)$ were studied:

$$g(\tau, m) = g_1(\tau) g_2(m), \qquad (5)$$

$$g(\tau, m) = g_1(\tau) g_3(f_{\tau,m}). \qquad (6)$$

[2] In Sect. 2.4 where the fast algorithm is presented, the sampling of τ has only minor effect on computational effi ciency, and therefore very dense sampling can be implemented. In practice, $\Delta\tau = 0.5$ suffi ces.

Reducing the two-parameter function $g(\tau, m)$ to a product of two marginal functions makes the optimization task easier and is likely to lead to a result that generalizes better to previously unseen test cases.

Let us first consider the form given by (5). The function $g_1(\tau)$ was parametrized by interpolating between ten "anchor points" which were distributed roughly as a geometric series between the fundamental frequencies 30 Hz and 2500 Hz. Similarly, the function $g_2(m)$ was parametrized by distributing ten anchor points as a geometric series between the 1st and 21st harmonic, and the function $g_2(m)$ was then linearly interpolated between these. The optimization was done by initializing the amplitudes of the anchor points to unity values and then updating them cyclically, one at the time, so as to minimize the F0 estimation error rate.

Figure 2 shows the learned functions $g_1(\tau)$ and $g_2(m)$, together with the resulting F0 estimation error rates (for the training data). The found shape of $g_1(\tau)$ is more or less a linear function of F0 (that is, f_s/τ), whereas $g_2(m)$ converged roughly to the $1/m$ shape, however with smaller weights for the lowest even-numbered harmonics (see Fig. 2). The predominant-F0 estimation accuracy is good, but multiple-F0 estimation leaves room for improvement.

Optimization for the factorization in (6) was done in a similar manner. The function $g_3(f_{\tau,m})$ was parametrized by interpolating it linearly between 13 anchor points that were distributed roughly as a geometric series between 30 Hz and 7 kHz. The optimization was again carried out by updating the amplitudes of the anchor points cyclically, one at a time, so as to minimize the F0 estimation error rate. The best result was obtained by starting the optimization from configuration $g(\tau, m) = 1/m$, which is obtained by initializing the anchor points as $g_1(\tau) = f_\mathrm{s}/\tau$ and $g_3(f_{\tau,m}) = 1/f_{\tau,m}$.

Figure 3 shows the learned functions $g_1(\tau)$ and $g_3(f_{\tau,m})$, and the resulting F0 estimation error rates. As can be seen, the functions $g_1(\tau)$ and $g_3(f_{\tau,m})$ do not drift very far from their initial shape, and the error rates are about the same as those achieved with the previous factorization. The latter form (6) is interesting, because it allows a simple implementation where the spectrum $Y(k)$ is first filtered using the response $g_3(f_{\tau,m})$, then $\hat{s}(\tau)$ is computed without any weights, and in the end $\hat{s}(\tau)$ is weighted with $g_1(\tau)$.

The latter factorization (6) was taken into use. To get rid of the large number free parameters (the anchor points), the function $g_1(\tau)$ is modeled as a linear function of fundamental frequency, $g_1(\tau) = f_\mathrm{s}/\tau + \alpha$, and the function $g_3(f_{\tau,m})$ is modeled as an inverse of the frequency $f_{\tau,m}$ and a moderation term β, $g_3(f_{\tau,m}) = 1/(f_{\tau,m} + \beta)$. The dashed lines in Figure 3 show the modeled functions. As a result, the function $g(\tau, m)$ can be finally written as

$$g(\tau, m) = \frac{f_\mathrm{s}/\tau + \alpha}{m f_\mathrm{s}/\tau + \beta}, \tag{7}$$

where the parameters α and β are given in Sect. 3.

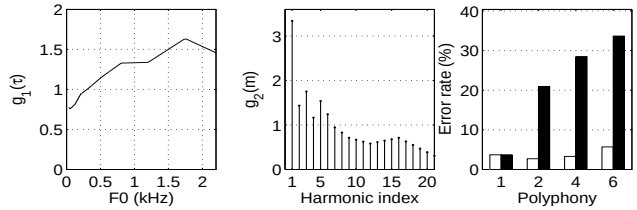

Figure 2. The learned functions $g_1(\tau)$ and $g_2(m)$ are shown in the left and the middle panels, respectively. For clarity, $g_1(\tau)$ is drawn as a function of F0 (f_s/τ) instead of τ. The right panel shows the resulting error rates for multiple-F0 estimation (black) and predominant-F0 estimation (white).

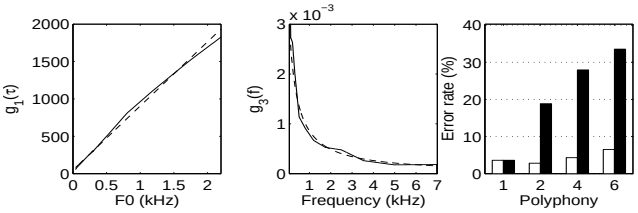

Figure 3. The learned functions $g_1(\tau)$ and $g_3(f_{\tau,m})$ are drawn with a solid line in the left and the middle panels, respectively. The dashed lines show the corresponding parametric models. The right panel shows F0 estimation error rates before the parametric modeling.

2.4. Iterative estimation and cancellation

The "direct" F0 estimator described above suffers from the problem that a single F0 in a sound mixture produces several peaks to $\hat{s}(\tau)$, and although the maximum of $\hat{s}(\tau)$ is a robust indicator of one of the true F0s, the second or third-highest peak is often due to the same sound and located at τ that is half or twice the position of the highest peak.

Multiple-F0 estimation accuracy can be improved by an iterative estimation and cancellation scheme where each detected sound is cancelled from the mixture and $\hat{s}(\tau)$ is updated accordingly before estimating the next F0. The basic cancellation mechanism described here is similar to that presented in [5], except that here a fast algorithm is described for finding the maximum of $\hat{s}(\tau)$ and a technique is proposed for estimating the number of sounds in the mixture.

Let us first look at an efficient way of finding the maximum of $\hat{s}(\tau)$. Somewhat surprisingly, the global maximum of $\hat{s}(\tau)$ and the corresponding value of τ can be found with a fast algorithm that does not require evaluating $\hat{s}(\tau)$ for all τ. This is another motivation for the iterative estimation and cancellation approach where only the maximum of $\hat{s}(\tau)$ is needed at each iteration.

Let us denote the minimum and maximum fundamental period of interest by τ_min and τ_max, respectively, and the required precision of sampling τ by τ_prec. A fast search of the maximum of $\hat{s}(\tau)$ can be implemented by repeatedly splitting the range $[\tau_\mathrm{min}, \tau_\mathrm{max}]$ into smaller "blocks", computing an upper bound for the salience within each block q, $s_\mathrm{max}(q)$, and continuing by splitting the block with the

Algorithm 1: Fast search of the maximum of $\hat{s}(\tau)$

1 $Q \leftarrow 1$; $\tau_{\mathrm{low}}(1) \leftarrow \tau_{\min}$; $\tau_{\mathrm{up}}(1) \leftarrow \tau_{\max}$; $q_{\mathrm{best}} \leftarrow 1$;
2 **while** $\tau_{\mathrm{up}}(q_{\mathrm{best}}) - \tau_{\mathrm{low}}(q_{\mathrm{best}}) > \tau_{\mathrm{prec}}$ **do**
3 # Split the best block and compute new limits
4 $Q \leftarrow Q + 1$
5 $\tau_{\mathrm{low}}(Q) \leftarrow (\tau_{\mathrm{low}}(q_{\mathrm{best}}) + \tau_{\mathrm{up}}(q_{\mathrm{best}}))/2$
6 $\tau_{\mathrm{up}}(Q) \leftarrow \tau_{\mathrm{up}}(q_{\mathrm{best}})$
7 $\tau_{\mathrm{up}}(q_{\mathrm{best}}) \leftarrow \tau_{\mathrm{low}}(Q)$
8 # Compute new saliences for the two block-halves
9 **for** $q \in \{q_{\mathrm{best}}, Q\}$ **do**
10 Calculate $s_{\max}(q)$ using Equations (3)-(4)
  with $g(\tau, m) = \frac{f_{\mathrm{s}}/\tau_{\mathrm{low}}(q)+\alpha}{m f_{\mathrm{s}}/\tau_{\mathrm{up}}(q)+\beta}$
  $\tau = (\tau_{\mathrm{low}}(q) + \tau_{\mathrm{up}}(q))/2$
11 $\Delta\tau = \tau_{\mathrm{up}}(q) - \tau_{\mathrm{low}}(q)$
12 **end**
13 # Search again the best block
14 $q_{\mathrm{best}} \leftarrow \arg\max_{q \in [1,Q]} s_{\max}(q)$
15 **end**
  Return $\hat{\tau} = (\tau_{\mathrm{low}}(q_{\mathrm{best}}) + \tau_{\mathrm{up}}(q_{\mathrm{best}}))/2$
16 $\hat{s}(\hat{\tau}) = s_{\max}(q_{\mathrm{best}})$

highest $s_{\max}(q)$. Let us denote the number of blocks by Q and the upper and lower limits of block q by $\tau_{\mathrm{low}}(q)$ and $\tau_{\mathrm{up}}(q)$, respectively. Index of the highest-salience block is denoted by q_{best}. The algorithm starts with only one block with upper and lower limits at $\tau_{\min}$ and $\tau_{\max}$, and then repeatedly splits the best block into two halves, as detailed in Algorithm 1.[3] As a result, it gives the maximum of $\hat{s}(\tau)$ and the corresponding value of τ.

On lines 13–14 of the algorithm, in order to obtain an upper bound for the salience $\hat{s}(\tau)$ within range $[\tau_{\mathrm{low}}(q), \tau_{\mathrm{up}}(q)]$, Equation (3) is evaluated using the given values for $g(\tau, m)$, τ, and $\Delta\tau$. Splitting a block later on can only decrease the value of $s_{\max}(q)$ when computed for the new block-halves.

Algorithm 1 is important for two reasons. First, it allows searching the maximum of $\hat{s}(\tau)$ efficiently even when the required sampling density of τ is very high. Secondly, increasing the sampling density of τ has the consequence that all the sets $\kappa_{\tau,m}$ in (3) contain exactly one frequency bin, in which case the non-linear maximization operation vanishes and $\hat{s}(\tau)$ becomes a linear function of the magnitude spectrum $|Y(k)|$, making it analytically more tractable.

The iterative estimation and cancellation goes as follows:

1. A residual spectrum $Y_{\mathrm{R}}(k)$ is initialized to equal $Y(k)$, and a spectrum of detected sounds $Y_{\mathrm{D}}(k)$ to zero.
2. A fundamental period $\hat{\tau}$ is estimated using $Y_{\mathrm{R}}(k)$ and Algorithm 1. The maximum of $\hat{s}(\tau)$ determines $\hat{\tau}$.
3. Harmonic partials of $\hat{\tau}$ are located in $Y_{\mathrm{R}}(k)$ at bin $\langle mK/\tau \rangle$. The frequency and amplitude of each partial is estimated and used to calculate its magnitude spectrum at the few surrounding frequency bins. The magnitude spectrum of the m:th partial is weighted by $g(\hat{\tau}, m)$ and added to the corresponding position of the spectrum of detected sounds, $Y_{\mathrm{D}}(k)$.
4. The residual spectrum is recalculated as $Y_{\mathrm{R}}(k) \leftarrow \max(0, Y(k) - dY_{\mathrm{D}}(k))$, where d controls the amount of the subtraction.
5. If there are sounds remaining in $Y_{\mathrm{R}}(k)$, return to Step 2.

Note that the purpose of the cancellation is to suppress harmonic and subharmonic peaks of $\hat{\tau}$ in $\hat{s}(\tau)$. This should be done in such a way that the residual spectrum $Y_{\mathrm{R}}(k)$ is not corrupted too much to detect remaining sounds at the coming iterations. These conflicting requirements are effectively met by weighting the partials of a detected sound by $g(\tau, m)$ in Step 3 before adding them to $Y_{\mathrm{D}}(k)$. In practice this means that the higher partials are not entirely cancelled from the mixture since $g(\tau, m) \approx 1/m$. Parameter $d \approx 1$ together with $g(\tau, m)$ defines the amount of subtraction.

The function $g(\tau, m)$ was re-optimized for the iterative method using a similar optimization scheme as described above. Despite the double role of $g(\tau, m)$ here (affecting both the salience and the cancellation), the obtained functions $g_1(\tau)$, $g_2(m)$, and $g_3(f_{\tau,m})$ were very similar to those shown in Figs. 2–3, and the model (7) is suitable.

When the number of sounds in the mixture is not given, it has to be estimated. This task, *polyphony estimation*, is accomplished by stopping the iteration when a newly-detected sound $\hat{\tau}_j$ at iteration j no longer increases the quantity

$$S(j) = \frac{\sum_{i=1}^{j} \hat{s}(\hat{\tau}_i)}{j^{\gamma}}, \tag{8}$$

where $\gamma = 0.70$ was found empirically.[4] The value of j maximizing (8) is taken as the estimated polyphony $\hat{P}$.

2.5. Joint estimation of multiple F0s

The described iterative multiple-F0 estimator is efficient and produces good results, but it also leaves us with a couple of open questions. How much does the iterative search algorithm affect the result? Is it possible to compute saliences of the found sounds so that the order of detecting them would not affect? This section describes a joint estimator which can answer these questions.

First, the salience function $\hat{s}(\tau)$ is calculated according to (3). Then, I highest local maxima of $\hat{s}(\tau)$ are chosen as candidate fundamental period values τ_i, $i = 1, \ldots, I$. For each candidate i, the following quantities are computed:

- Frequency bins of harmonic partials $k_{i,m}$, where m is the harmonic index and $k_{i,m}$ corresponds to the maximum of $|Y(k)|$ in the range κ_{m,τ_i} (see (4)).

[3] It is even more effi cient to start with $\sqrt{(\tau_{\max} - \tau_{\min})/\tau_{\mathrm{prec}}}$ blocks.

[4] Note that $S(j)$ would be monotonically decreasing for $\gamma = 1$ (average of $\hat{s}(\hat{\tau}_i)$:s) and monotonically increasing for $\gamma = 0$ (sum).

- Candidate spectrum $Z_i(k)$ is an estimate of the spectrum of candidate i, and is calculated by translating the spectrum of the window function (Hanning) to the positions $k_{i,m}$ and adding them to $Z_i(k)$ after scaling by $(d/2)g(\tau_i, m)$, where d is the cancellation parameter from Step 4 of the iterative method.

Let us denote by P the number of simultaneous F0s to estimate and by $\mathcal{I}$ a set of P different candidate indices i (there are $\binom{I}{P}$ different possibilities). Then the joint estimation consists of finding such an index set $\mathcal{I}$ that maximizes

$$G(\mathcal{I}) = \sum_{i\in\mathcal{I}} \sum_m g(\tau_i, m)|Y(k_{i,m})| \prod_{j\in\mathcal{I}\setminus i} (1 - Z_j(k_{i,m})), \tag{9}$$

where $Z_j(k) \leq 1$ because $(d/2)g(\tau, m) \leq 1$. By comparison with (1), it can be seen that the above goodness measure implements a similar harmonic summing model but with the difference that the salience contribution of sound i is reduced by "inhibition" (cancellation) from other sounds j in $\mathcal{I}$, as determined by their estimated spectrum $Z_j(k)$. In fact, the above model is a very close equivalent to the iterative method presented above, the difference being that here the estimation is performed jointly instead of iteratively. The reason why the parameter d is halved when calculating $Z_j(k)$ is that here all sounds inhibit all others, whereas in the iterative method only sounds detected at earlier iterations inhibit (through cancellation) those detected later.

A problem with (9) is that the computational complexity of evaluating $G(\mathcal{I})$ for all $\binom{I}{P}$ different index combinations $\mathcal{I}$ is computationally impractical. A reasonably efficient implementation is possible by making use of the lower bound $\tilde{G}(\mathcal{I})$ of $G(\mathcal{I})$. By writing out the product in (9), it is easy to see that $G(\mathcal{I}) \geq \tilde{G}(\mathcal{I})$ where

$$\begin{aligned}\tilde{G}(\mathcal{I}) &= \sum_{i\in\mathcal{I}} \sum_m g(\tau_i, m)|Y(k_{i,m})|[1 - \sum_{j\in\mathcal{I}\setminus i} Z_j(k_{i,m})] \\ &= \sum_{i\in\mathcal{I}} \hat{s}(\tau_i) - \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}\setminus i} \mathrm{Inh}(i, j)\end{aligned} \tag{10}$$

where the "inhibition" $\mathrm{Inh}(i, j)$ is a non-symmetric function

$$\mathrm{Inh}(i, j) = \sum_m g(\tau_i, m)|Y(k_{i,m})|Z_j(k_{i,m}). \tag{11}$$

From the computational viewpoint, the advantage of (10) is that the values $\hat{s}(\tau_i)$ and $\mathrm{Inh}(i, j)$ can be precomputed, making the evaluation of (10) an easy operation for different index combinations $\mathcal{I}$. Another crucial factor is that, due to the sparseness of $Z_j(k)$, the lower bound $\tilde{G}(\mathcal{I}_c)$ is actually an accurate estimate of $G(\mathcal{I}_c)$ so that $G(\mathcal{I}_c) \approx \tilde{G}(\mathcal{I}_c)$. The algorithm for finding a set $\mathcal{I}$ which maximizes (9) is:

1. Initialize I different sets $\mathcal{I}$ with the individual candidates τ_i, $i = 1, \ldots, I$, so that set number c is initialized with $\mathcal{I}_c \leftarrow \{c\}$ and $\tilde{G}(\mathcal{I}_c) \leftarrow \hat{s}(\tau_c)$.
2. Generate $I \times I$ new combinations by extending all the existing combinations with all the individual candidates τ_i. The goodness measures of these extended combinations can be computed recursively as
$$\tilde{G}(\mathcal{I}_c \cup i) = \tilde{G}(\mathcal{I}_c) + \hat{s}(\tau_i) - \sum_{j\in\mathcal{I}_c} (\mathrm{Inh}(i, j) + \mathrm{Inh}(j, i)).$$
3. Sort the $I \times I$ extended sets in descending goodness order and retain only I best combinations with different goodness measures. The latter prevents from choosing different permutations of a same set.
4. If polyphony P was not reached, return to Step 2.
5. Evaluate the exact goodness measure (9) for the I best combinations and choose the combination with the highest value to output.

If the polyphony P is not given, it can be estimated by evaluating the exact goodness measure (9) for the I best combination always between the Steps 3 and 4, and by storing the best j-size combination $\mathcal{I}_{\mathrm{best}(j)}$. Extending the sets is continued as long as the following measure increases:

$$S(j) = G(\mathcal{I}_{\mathrm{best}(j)})/j^\gamma, \tag{12}$$

where the parameter $\gamma = 0.73$ was found empirically.

An advantage of the joint estimation method is that, contrary to the iterative system, here the order of detecting the sounds does not affect the result. A drawback is that the joint estimator is computationally less efficient as it requires evaluating the function $\hat{s}(\tau)$ for all τ and therefore Algorithm 1 cannot be used. Finding the optimal combination $\mathcal{I}$ in the above algorithm is still quite efficient when 50–100 candidates τ_i are selected from $\hat{s}(\tau)$, which is roughly the amount needed to retain the true periods among them. In the simulations, we used $I = 100$.

3. Evaluation

Simulation experiments were carried out to evaluate the proposed estimators. These were compared with the reference methods [3] and [5] based on auditory models, and the reference method [4] based on spectral techniques. Test data consisted of random mixtures of musical instrument samples with F0s between 40 and 2100 Hz, generated in the same way as in Sect. 2.3 but of course randomizing new test cases here. [5] The acoustic database, however, was the same, and consisted of samples from the McGill University Master Samples collection, the University of Iowa website, IRCAM Studio Online, and of recordings for the acoustic guitar. In total, there were 2842 samples from 32 musical instruments.

As estimation of the number of concurrent sounds is very difficult in itself, we evaluate F0 estimation and polyphony estimation separately. The parameter values α, β,and d were the same for all the three proposed methods and were 27 Hz,

[5] For the reference method [3], F0s were restricted between 40 Hz and 530 Hz, since the method is not very robust for F0s higher than this.

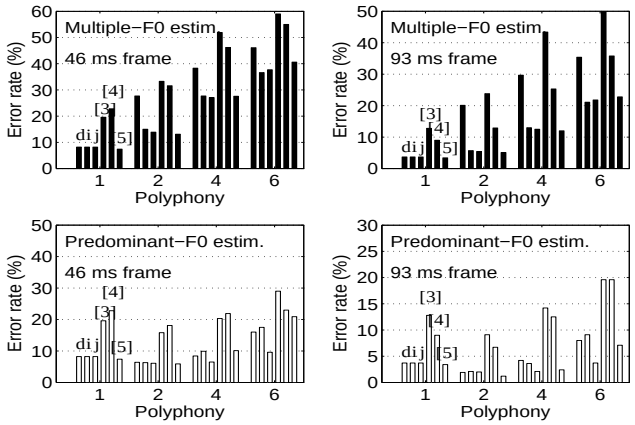

Figure 4. Multiple-F0 estimation and predominant-F0 estimation results in 46 ms and 93 ms analysis frames. Reading left to right, each stack of six thin bars corresponds to the error rates of the direct (d), iterative (i), joint (j), and reference methods [3], [4], and [5] in a certain polyphony.

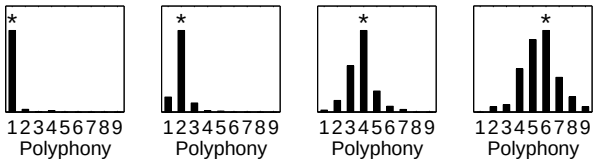

Figure 5. Histograms of polyphony estimates for the iterative method and a 93 ms analysis frame. The asterisks indicate the true polyphony (1, 2, 4, and 6, from left to right).

320 Hz, and 1.0, respectively, for 46 ms analysis frame, and 52 Hz, 320 Hz, and 0.89, respectively, for 93 ms frame.

Figure 4 shows the F0 estimation results of the proposed and the reference methods. Here the number of concurrent sounds (polyphony) was given as a side-information to the estimators. The error rates are practically the same for the proposed iterative and joint methods and the reference method [5], and these three outperform the methods [3] and [4]. This is a very nice result, since the best reference method [5] involves computation of an auditory model, including e.g. Fourier transforms at 70 subbands. The proposed methods are considerably simpler and computationally more efficient. In monophonic cases (polyph. 1), about 50% of the errors are caused by F0s between 40 and 65 Hz.

The lower panels of Figure 4 show predominant-F0 estimation accuracies. Here the error rates are practically the same for the proposed direct and the iterative method and for the reference method. The accuracy of the joint method, however, is clearly better in high polyphonies.

Figure 5 illustrates the results of polyphony estimation for the iterative method and a 93 ms analysis frame. Results for the joint method were very similar and are not shown. The asterisk indicates true polyphony in each panel, and bars show a histogram of the estimates. The results are not fully satisfactory, and it seems that robust estimation of the number of sounds requires more than one analysis frame.

4. Conclusions

The principle of summing harmonic amplitudes as given by (1) is very simple, yet it suffices for predominant-F0 estimation in polyphonic signals provided that the weights $g(\tau, m)$ of different partials and periods are appropriate. In multiple-F0 estimation, both the iterative and the joint estimator were successful, but the iterative method admits a fast implementation and is therefore more appealing. The joint estimator, in turn, achieves better predominant-F0 estimation. Both methods can be seen to implement the model embodied in the goodness measure (9), which is very simplistic considering the wide range of instruments and F0 values addressed.

References

[1] K. Kashino and H. Tanaka, "A sound source separation system with the ability of automatic tone modeling," in *International Computer Music Conf.*, (Tokyo, Japan), 1993.

[2] A. de Cheveigné, "Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model for auditory processing," *Journal of the Acoust. Soc. Am.*, vol. 93, no. 6, 1993.

[3] T. Tolonen and M. Karjalainen, "A computationally effi cient multipitch analysis model," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 6, pp. 708–716, 2000.

[4] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, 2003.

[5] A. P. Klapuri, "A perceptually motivated multiple-F0 estimation method for polyphonic music signals," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), 2005.

[6] M. Goto, "A real-time music scene description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[7] M. Davy, S. Godsill, and J. Idier, "Bayesian analysis of Western tonal music," *Journal of the Acoustical Society of America*, vol. 119, no. 4, pp. 2498–2517, 2006.

[8] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), 2003.

[9] S. A. Abdallah and M. D. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra," in *International Conference on Music Information Retrieval*, (Barcelona, Spain), pp. 318–325, Oct. 2004.

[10] G. E. Poliner and D. P. W. Ellis, "A classifi cation approach to melody transcription," in *6th International Conference on Music Information Retrieval*, (London, UK), 2005.

[11] W. J. Hess, *Pitch Determination of Speech Signals*. Berlin Heidelberg: Springer, 1983.

[12] P. Walmsley, *Signal Separation of Musical Instruments. Simulation-based methods for musical signal decomposition and transcription*. PhD thesis, Department of Engineering, University of Cambridge, Sept. 2000.