Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness

Anssi P. Klapuri

***Abstract*—A new method for estimating the fundamental frequencies of concurrent musical sounds is described. The method is based on an iterative approach, where the fundamental frequency of the most prominent sound is estimated, the sound is subtracted from the mixture, and the process is repeated for the residual signal. For the estimation stage, an algorithm is proposed which utilizes the frequency relationships of simultaneous spectral components, without assuming ideal harmonicity. For the subtraction stage, the spectral smoothness principle is proposed as an efficient new mechanism in estimating the spectral envelopes of detected sounds. With these techniques, multiple fundamental frequency estimation can be performed quite accurately in a single time frame, without the use of long-term temporal features. The experimental data comprised recorded samples of 30 musical instruments from four different sources. Multiple fundamental frequency estimation was performed for random sound source and pitch combinations. Error rates for mixtures ranging from one to six simultaneous sounds were 1.8%, 3.9%, 6.3%, 9.9%, 14%, and 18%, respectively. In musical interval and chord identification tasks, the algorithm outperformed the average of ten trained musicians. The method works robustly in noise, and is able to handle sounds that exhibit inharmonicities. The inharmonicity factor and spectral envelope of each sound is estimated along with the fundamental frequency.**

***Index Terms*—Acoustic signal analysis, fundamental frequency estimation, music, music transcription, pitch perception.**

I. Introduction

PITCH perception plays an important part in human hearing and understanding of sounds. In an acoustic environment, human listeners are able to perceive the pitches of several simultaneous sounds and make efficient use of the pitch to acoustically separate a sound in a mixture [1]. Computational methods for multiple fundamental frequency (F0) estimation have received less attention, though many algorithms are available for estimating the F0 in single-voice speech signals [2]–[4]. It is generally admitted that these algorithms are not appropriate as such for the multiple-F0 case.

A sound has a certain pitch if it can be reliably matched by adjusting the frequency of a sine wave of arbitrary amplitude [5]. Pitch is a perceptual attribute of sounds. The corresponding physical term F0 is defined for periodic or nearly periodic sounds only. For these classes of sounds, F0 is closely related to pitch and is defined as the inverse of the period, i.e., the time shift for which the time-domain signal shows high correlation with itself. In cases where the fundamental period is ambigous, a candidate closest to the subjective pitch period is regarded as the correct F0.

Musical signals are natural candidates for the problem of multiple-F0 estimation, in the same way as speech signals are natural candidates for single-F0 estimation. Automatic transcription of music aims at extracting the pitches, onset times, and durations of the notes that constitute the piece. The first multiple-F0 algorithms were designed for the purpose of transcribing polyphonic music in which several sounds are playing simultaneously. These attempts date back to 1970s, when Moorer built a system for transcribing duets, i.e., two-voice compositions [6]. The work was continued by Chafe and his collegues [7]. Further advances were made by Maher [8]. However, the early systems suffered from severe limitations in regard to the pitch ranges and relationships of simultaneous sounds, and the polyphony was restricted to two concurrent sounds. Relaxation of these constraints was attempted by allowing some more errors to occur in the transcription [9], or by limitation to one carefully modeled instrument [10], [11].

More recent transcription systems have recruited psychoacoustically motivated analysis principles, used sophisticated processing architectures, and extended the application area to computational auditory scene analysis in general [12]. Kashino *et al.* integrated signal analysis with temporal and musical predictions by applying a Bayesian probability network [13]. Martin utilized musical rules in transcribing four-voice piano compositions [14]. Front-end processing in his system was performed using a log-lag correlogram model of the human auditory periphery, as described in [15]. Goto was the first to introduce a system which works reasonably accurately for real-world complex musical signals by finding the melody and bass lines in them [16].

Multiple-F0 estimation is closely related to auditory scene analysis: any algorithm that can find the F0 of a sound and not get confused by other co-occurring sounds is, in effect, doing auditory scene analysis [1, p. 240]. Because the human auditory system is very accurate in performing this task, imitation of its processing principles has become common and psychoacoustically inspired systems in general have been relatively successful. Brown and Cooke have built computational models of the human auditory processes and also addressed the auditory grouping and streaming of musical sounds according to common acoustic properties [17]. Godsmark and Brown proposed a blackboard architecture to integrate evidence from different auditory organization principles and demonstrated

Manuscript received November 29, 2001; revised April 16, 2003. This work was supported by the TISE Graduate School, Tampere University of Technology, the Foundation of Emil Aaltonen, Tekniikan edistämissäätiö, and the Nokia Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sean A. Ramprashad.

The author is with Institute of Signal Processing, Tampere University of Technology, FIN-33720 Tampere, Finland (e-mail: klap@cs.tut.fi).

Digital Object Identifier 10.1109/TSA.2003.815516

that the model could segregate melodic lines from polyphonic music [18].

The unitary model of pitch perception proposed by Meddis and Hewitt has had a strong influence on F0 estimation research [19], [20]. Tolonen and Karjalainen have suggested a simplified version of the unitary pitch model and applied it to the multiple-F0 estimation of musical sounds [21]. In [22], de Cheveigné and Kawahara integrated the model with a concurrent vowel identification model of Meddis and Hewitt [23] and developed an approach where F0 estimation is followed by the cancellation of the detected sound and iterative estimation for the residual signal. A more straightforward version of this iterative approach was earlier proposed by de Cheveigné in [24].

The periodicity transform method proposed by Sethares and Staley in [25] bears a close resemblance to that of de Cheveigne in [24], although the former is purely mathematically formulated. A more dynamic approach to residue-driven processing has been taken by Nakatani and Okuno [26]. Their system was designed to segregate continuous streams of harmonic sounds, such as the voiced sections of two or three simultaneous speakers. Multiple agents were deployed to trace harmonic sounds in stereophonic input signals, the sounds were subtracted from the input signal, and the residual was used to update the parameters of each sound and to create new agents when new sounds were detected.

There are two basic problems that a multiple-F0 estimator has to solve in addition to those that are confronted with in single-F0 estimation. First, the calculated likelihoods (or weights) of different F0 candidates must not be too much affected by the presence of other, co-occurring sounds. To achieve this, multiple-F0 algorithms typically decompose incoming signals into smaller elements which are then selectively used to calculate the weight for each candidate. For example, some methods trace sinusoidal components and then group them into sound sources according to their individual attributes, such as harmonic relationships or synchronous changes in the components [7], [13], [16], [26], [27]. Other algorithms apply comb filtering in the time domain to select only the harmonically related components [22], [24], [25]. Several recent systems have employed auditory models which break an incoming sound into subchannel signals and perform periodicity analysis withing channels [18], [20], [22].

In the second place, even when a correct F0 has been detected, the next-highest weights are often assigned to half or twice of this correct F0 value. Thus, the effect of any detected F0 must be cancelled from harmonics and subharmonics before deciding the next most likely F0. Some algorithms perform this by manipulating the calculated F0 weights directly [21]. Other methods estimate the spectrum of each detected sound and then subtract it from the mixture in an iterative fashion [24], [25], or process as a joint estimation and cancellation pursuit [24], [26]. The latter scheme is similar to the analysis-by-synthesis techniques in parametric coding, where for example sinusoidal components are detected, modeled, and subtracted from the input in order to minimize the residual signal [28].

The aim of this paper is to propose a multiple-F0 analysis method that operates at the level of a single time frame and is applicable for sound sources of diverse kinds. Automatic transcription of music is seen as an important application area, implying a wide pitch range, varying tone colors, and a particular need for robustness in the presence of other harmonic and noisy sounds.

An overview of the proposed system is illustrated in Fig. 1. The method operates iteratively by estimating and removing the most prominent F0 from the mixture signal. The term *predominant-F0 estimation* refers to a crucial stage where the F0 of the most prominent sound is estimated in the presence of other harmonic and noisy sounds. To achieve this, the harmonic frequency relationships of simultaneous spectral components are used to group them to sound sources. An algorithm is proposed which is able to handle inharmonic sounds. These are sounds for which the frequencies of the overtone partials (harmonics) are not in exact integer ratios. In a subsequent stage, the spectrum of the detected sound is estimated and subtracted from the mixture. This stage utilizes the spectral smoothness principle, which refers to the expectation that the spectral envelopes of real sounds tend to be slowly varying as a function of frequency. In other words, the amplitude of a harmonic partial is usually close to the amplitudes of the nearby partials of the same sound. The estimation and subtraction steps are then repeated for the residual signal. A review and discussion of the earlier iterative approaches to multiple-F0 estimation can be found in [22], [24]. Psychoacoustic evidence in favor of the iterative approach can be found in [1, p. 240, 244], [5].

The motivation for this work is in practical engineering applications, although psychoacoustics is seen as an essential base of the analysis principles. The proposed algorithm is able to resolve at least a couple of the most prominent F0s, even in rich polyphonies. Reliable estimation can be carried out in cases where the signal has been corrupted by high levels of additive noise or where wide frequency bands are missing. Non-ideal sounds that exhibit inharmonicities can be handled. The applications thus facilitated comprise transcription tools for musicians, transmission and storage of music in a compact form, and new ways of searching musical information.

The paper is organized as follows. Section II will describe the different elements of the algorithm presented in Fig. 1. These include preprocessing, the harmonicity principle used, the smoothing of detected sounds, and estimation of the number of concurrent sounds. Section III will describe experimental results and will compare these with the performance of two reference methods and human listeners. Finally, Section IV will summarize the main conclusions and will discuss future work.

II. Proposed Multiple-F0 Estimation Method

This section will look at all the necessary elements required for the multiple-F0 estimation task and as illustrated in Fig. 1. To begin, Section II-A will describe the preprocessing stage which is necessary to achieve robustness in additive noise and to handle sounds with uneven spectral shapes. Next, the main principle behind using harmonic relationships is discussed in Section II-B. Section II-C will describe the smoothing algorithm which is needed to subtract each detected sound from the mixture so that the remaining sounds are not corrupted. The last subsection will propose a mechanism to control the stopping of the iterative estimation and cancellation process.

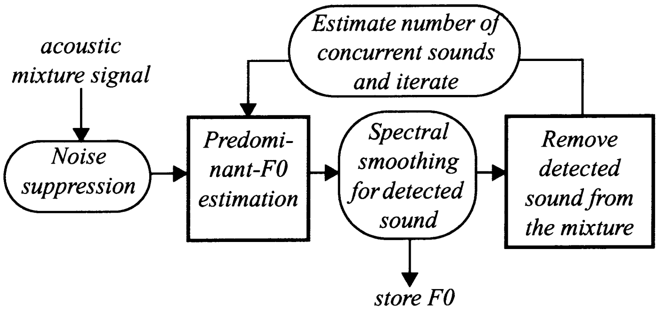
acoustic mixture signal
Noise suppression
Predominant-F0 estimation
Estimate number of concurrent sounds and iterate
Spectral smoothing for detected sound
Remove detected sound from the mixture
store F0

Fig. 1. Overview of the proposed multiple-F0 estimation method.

A. Preprocessing

All calculations in the proposed system take place in the frequency domain. A discrete Fourier transform is calculated for a Hamming-windowed frame of an acoustic input signal, sampled at 44.1 kHz rate and quantized to 16-bit precision. Frame lengths of 93 ms and 190 ms were used in simulations. These may seem long from the speech processing point of view, but are actually not very long for musical chord identification tasks. In such tasks, the pitch range is wide, mixtures of low sounds produce very dense sets of frequency partials, and F0 precision of 3% is required to distinguish adjacent notes (see Appendix).

Preprocessing the spectrum before the actual multiple-F0 analysis is an important factor in the performance of the system. It provides robustness in additive noise and ensures that sounds with varying spectral shapes can be handled. The signal model assumed by the proposed system is

$$X(k) = H(k)S(k) + N(k) \tag{1}$$

where $X(k)$ is the discrete power spectrum of an incoming acoustic signal and $S(k)$ is the power spectrum of a vibrating system whose fundamental frequency should be measured. The factor $H(k)$ represents the frequency response of the operating environment and the body of a musical instrument which filters the signal of the vibrating source. Elimination of $H(k)$ is often referred to as pre-whitening. The term $N(k)$ represents the power spectrum of additive noise. In music signals, the additive interference is mainly due to the transient-like sounds of drums and percussive instruments.

In principle, additive noise can be suppressed by performing spectral subtraction in the power spectral domain. The effect of $H(k)$, in turn, can be suppressed by highpass liftering[1] the log-magnitude spectrum. Confirming the reports of earlier authors, however, two noise-reduction systems in a cascade does not produce appropriate results [30]. Rather, successful noise suppression is achieved by applying magnitude warping which equalizes $H(k)$ while allowing the additive noise to be linearly subtracted from the result. The power spectrum $X(k)$ is magnitude-warped as

$$Y(k) = \ln\left\{1 + \frac{1}{g}X(k)\right\} \tag{2}$$

where

$$g = \left[\frac{1}{k_1 - k_0 + 1}\sum\nolimits_{l=k_0}^{k_1} X(l)^{\frac{1}{3}}\right]^3. \tag{3}$$

The frequency indices k_0 and k_1 correspond to frequencies 50 Hz and 6.0 kHz, respectively, and are determined by the frequency range utilized by the multiple-F0 estimator. The exact formula for calculating g is not as critical as the general idea represented by (2). The use of (2) and (3) is based on two reasonable assumptions. First, the amplitudes of the important frequency partials in $H(k)S(k)$ are above the additive noise $N(k)$. Secondly, it is assumed that a majority of the frequency components between k_0 and k_1 correspond to the additive noise floor, not to the spectral peaks of $H(k)S(k)$. In this case, $(1/g)$ scales the input spectrum so that the level of additive noise $N(k)$ stays close to unity and the spectral peaks of the vibrating system $H(k)S(k)$ are noticeably above unity. It follows that in (2), additive noise goes through a linear-like magnitude-warping transform, whereas spectral peaks go through a logarithmic-like transform.

The response $H(k)$ is efficiently flattened by the logarithmic-like transform, since subsequent processing takes place in the warped magnitude scale. Additive noise is suppressed by applying a specific spectral subtraction on $Y(k)$ [34]. A moving average $\hat{N}(k)$ over $Y(k)$ is calculated on a logarithmic frequency scale and then linearly subtracted from $Y(k)$. More exactly, local averages were calculated at 2/3-octave bands while constraining the minimum bandwidth to 100 Hz at the lowest bands. The same bandwidths are used in the subsequent F0 calculations and are motivated by the frequency resolution of the human auditory system and by practical experiments with generated mixtures of musical sounds and noise. The use of the logarithmic frequency scale was clearly advantageous over a linear scale since it balances the amount of spectral fine structure that is used with different F0s.

The estimated spectral average $\hat{N}(k)$ is linearly subtracted from $Y(k)$ and resulting negative values are constrained to zero

$$Z(k) = \max\left\{0,\ Y(k) - \hat{N}(k)\right\}. \tag{4}$$

The preprocessed spectrum $Z(k)$ is passed to the multiple-F0 estimator.

B. Harmonicity Principle

In this section, the "Predominant-F0 estimation" part of the algorithm is described. A process is proposed which organizes mixture spectra by utilizing the harmonic relationships between frequency components, without assuming ideal harmonicity.

Several fundamentally different approaches to F0 estimation have been proposed. One category of algorithms measures periodicity in the time-domain signal. These methods are typically based on calculating the time-domain autocorrelation function or the cepstrum representation [32], [33]. As shown in [34], this is theoretically equivalent to matching a pattern of frequency partials at *harmonic positions* of the sound spectrum. An explicit way of building upon this idea is to perform harmonic pattern matching in the frequency domain [35], [36]. Another category of algorithms measures periodicity in the frequency-domain, observing F0 from the *intervals* between the frequency partials of a sound. The spectrum autocorrelation method and its variants have been successfully used in several F0 estimators [37], [38]. An interesting difference

[1] The term "liftering" is defined [29].

between the time-domain and frequency-domain periodicity analysis methods is that the former methods are prone to errors in F0 halving and the latter to errors in F0 doubling. This is because the time-domain signal is periodic at half the F0 rate (twice the fundamental time delay) and the spectrum is periodic at double the F0 rate. A third, psychoacoustically motivated group of algorithms measures the *periodicity of the amplitude envelope* of a time-domain signal within several frequency channels [20], [21], [39].

A major shortcoming of many of the earlier proposed methods is that they do not handle inharmonic sounds appropriately. In the case of real nonideal physical vibrators, the harmonic partials are often not in exact integral ratios. For example for stretched strings the frequency f_h of an overtone partial h obeys

$$f_h = hF\sqrt{1+(h^2-1)\beta} \tag{5}$$

where F is the fundamental frequency and β is the inharmonicity factor [40]. Equation (5) means that the partials cannot be assumed to be found at harmonic spectrum positions, but are gradually shifted upwards in the spectrum. This is not of great concern in speech processing, but is important when analyzing musical sounds at a wide frequency band [41]. In the rest of this paper, capital letter F is used to denote fundamental frequency, and the lower case letter f to denote simply frequency.

The proposed predominant-F0 estimation method works by calculating independent F0 estimates at separate frequency bands and then combining the results to yield a global estimate. This helps to solve several difficulties, one of which is inharmonicity. According to (5), the higher harmonics may deviate from their expected spectral positions, and even the intervals between them are not constant. However, we can assume the spectral intervals to be piecewise constant at narrow-enough frequency bands. Thus, we utilize spectral intervals in a two step process which 1) calculates the weights of different F0s at separate frequency bands and 2) combines the results in a manner that takes inharmonicity into account. Another advantage of bandwise processing is that it provides robustness and flexibility in the case of badly corrupted signals where only a fragment of the whole frequency range can be used [41]. The two steps are now described.

1) Bandwise F0 Estimation: The preprocessed spectrum $Z(k)$ is analyzed at 18 bands that distribute approximately logarithmically between 50 Hz and 6 kHz, as illustrated in Fig. 2. Each band b comprises a 2/3-octave region of the spectrum, constraining, however, the minimum bandwidth to 100 Hz. Band b is subject to weighting with a triangular frequency response $G_b(k)$, shown in Fig. 2. The overlap between adjacent bands is 50%, making the overall response sum to unity at all except the lowest bands. Response at band b is denoted by

$$Z_b(k) = G_b(k)Z(k). \tag{6}$$

Non-zero frequency components of $Z_b(k)$ are defined for frequency indices, $k \in [k_b, k_b + K_b - 1]$ where k_b is the lowest frequency component at band b and K_b is the number of components at the band.

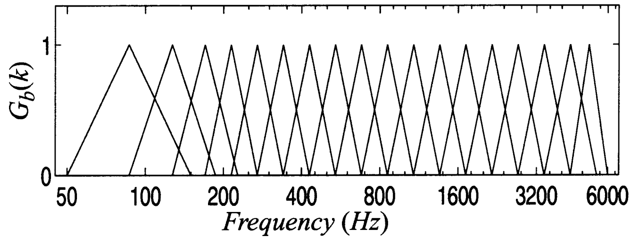

Fig. 2. Magnitude responses of the 18 frequency bands, at which the bandwise F0 estimation takes place.

In each band, the algorithm calculates a weight vector $L_b(n)$ across frequency indices. Note, index n corresponds to the fundamental frequency $F = (n/K)f_s$ where K is the number of samples in the time-domain analysis frame and f_s is the sampling rate. The resolution of the weight vector is the same as that of the preprocessed spectrum $Z(k)$. The bandwise weights $L_b(n)$ are calculated by finding a series of each n^{th} frequency components at band b that maximizes the sum

$$L_b(n) = \max_{m\in \boldsymbol{M}}\left\{c(m,n)\sum_{j=0}^{J(m,n)-1} Z_b(k_b+m+nj)\right\} \tag{7}$$

where

$$J(m,n) = \left\lceil \frac{(K_B-m)}{n} \right\rceil \tag{8}$$

$$c(m,n) = \left[\frac{0.75}{J(m,n)}\right] + 0.25. \tag{9}$$

Here, $\boldsymbol{M} = \{0, 1, \ldots, k-1\}$ is the offset of the series of partials in the sum, $J(m,n)$ is the number of partials in the sum, and $c(m,n)$ is a normalization factor. A normalization factor is needed because J varies for different values of m and n. The form $c(m,n)$ was determined by training with isolated musical instrument samples in varying noise conditions. The offset m is varied to find the maximum of (7), which is then stored in $L_b(n)$. Different offsets have to be tested because the series of higher harmonic partials may have shifted due to inharmonicity.

The upper panel in Fig. 3 illustrates the calculations for a single harmonic sound at the band $b = 12$ between 1100 Hz and 1700 Hz. The arrows indicate the series of frequency components which maximizes $L_{12}(n)$ for the true F0.

The values of the offset m are restricted to physically realistic inharmonicities, a subset of $\boldsymbol{M}$. The exact limit is not critical, therefore (5) with a constant $\beta = 0.01$ inharmonicity factor can be used to determine the maximum allowable offset from the ideal harmonic positions. The harmonic index h in (5) can be approximated by $h \approx (k_b + K_b - 1)/n$. It follows that the fundamental partial $h = 1$ must be exactly in the harmonic spectral position, whereas the whole set $\boldsymbol{M}$ has to be considered for the highest partials. In other words, the algorithm combines the use of spectral positions for the lowest harmonic partials and the use of spectral intervals for the higher partials. For a frequency band which is assumed to contain only the first harmonic partial of a sound with fundamental frequency corresponding to index n, inharmonicity is not allowed. Here J is set to 1, and (7) reduces to the special case

$$L_b(n) = Z_b(n). \tag{10}$$

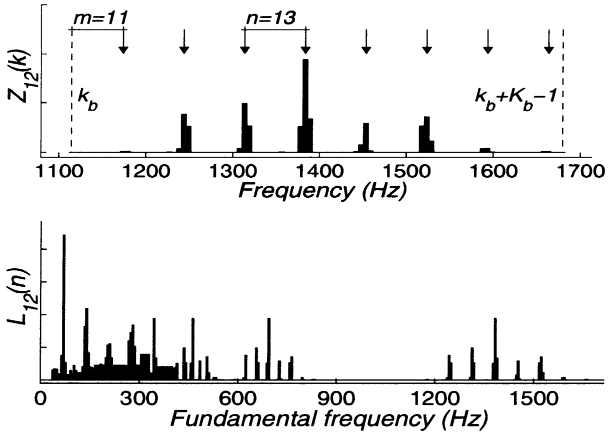

Fig. 3. Calculation of the bandwise F0 weight vectors according to (7).

It follows that in this case the weights $L_b(n)$ are equal to $Z_b(n)$ between the frequency limits of the band. The algorithm is detailed in Table I.

The lower panel in Fig. 3 shows the entire weight vector $L_{12}(n)$ calculated at band $b = 12$ for the same signal as in the upper panel. As can be seen, the preprocessed spectrum $Z_{12}(n)$ appears as such at the corresponding band of $L_{12}(n)$. A twice narrower copy of $Z_{12}(n)$ is found an octave below, since the F0s in that range have exactly one harmonic partial at the band (the second partial). Yet lower F0 candidates have a series of higher overtones at the band and inharmonicity is allowed. This is the case for the true F0 (70 Hz) which has been assigned the highest weight.

An important property of the presented calculations is that only the selected frequency samples contribute to the weight $L_b(n)$, not the overall spectrum. The other co-occurring sounds affect the weight only to the extent that their partials overlap those of the sound being estimated (a solution for overlapping partials is given in Section II-C). Harmonic selection provides robustness in sound mixtures as long as we do not rely on the detection of single partials, as is the case here. Harmonic selection was originally proposed by Parsons in [27] and is used in most multiple-F0 algorithms, as described in Section I.

2) Integration of Weights Across Subbands: Fig. 4 shows the calculated $L_b(n)$ weight vectors at different bands for two isolated piano tones where the weight vectors are arranged in increasing band center frequency order. As expected, the maximum weight is usually assigned to the true F0, provided that there is a harmonic partial at that band. The inharmonicity phenomenon appears in Figs. 4(a) and 4(b) as a rising trend in the fundamental frequency.

The bandwise F0 weights are combined to yield a global F0 estimate. A straightforward summation across the weight vectors does not accumulate them appropriately since the F0 estimates at different bands may not match for inharmonic sounds, as can be seen from Fig. 4. To overcome this, the inharmonicity factor is estimated and taken into account. Two different inharmonicity models were implemented, the one given in (5) and another mentioned in [40, p. 363]. In simulations, the performance difference between the two was negligible. The model in (5) was adopted.

Global weights $L(n)$ are obtained by summing squared bandwise weights $L_b(n)$ that are selected from different bands according to a curve determined by (5). A search over possible values of $\beta(n)$ is conducted for each n, and the highest $L(n)$ and the corresponding $\beta(n)$ are stored in the output. Squaring the bandwise F0 weights prior to summing was found to provide robustness in the presence of strong interference where the pitch may be perceptible only at a limited frequency range.

TABLE I
ALGORITHM FOR CALCULATING THE WEIGHTS $L_b(n)$ FOR DIFFERENT F0s AT BAND b. SEE TEXT FOR THE DEFINITION OF SYMBOLS

```
# Implementation of the model in Eq. (7)
$n_0 \leftarrow \mathrm{floor}[(F_{min}/f_s)K]$
$n_1 \leftarrow K_b - 1$
$l_b \leftarrow k_b + K_b - 1$
for $n \leftarrow$ from $n_0$ to $n_1$ do
    $m_0 = \mathrm{round}[\mathrm{ceil}(k_b/n)n] - k_b$
    $\delta \leftarrow \frac{l_b f_s}{K}[\sqrt{1 + 0.01[(l_b/n)^2 - 1]} - 1]$
    $m_1 \leftarrow m_0 + \delta$
    if $m_1 > m_0 + n - 1$ then
        $m_0 \leftarrow 0$
        $m_1 \leftarrow n - 1$
    $L_b(n) \leftarrow 0$
    for $m \leftarrow$ from $m_0$ to $m_1$ do
        $J \leftarrow \mathrm{floor}[(K_b - m - 1)/n] + 1$
        $L_{now} \leftarrow (0.75/J + 0.25) \times$
                $\sum_{j=0}^{J-1} Z_b(k_b + m + nj)$
        if $L_{now} > L_b(n)$ then
            $L_b(n) \leftarrow L_{now}$
    end
end
# Range of n that have exactly one harmonic partial
# at frequency band b (inharmonicity not allowed)
$h \leftarrow 1$
$k_0 \leftarrow \mathrm{floor}[(k_b + K_b)/(h + 1)]$
if $k_0 < k_b$ then $k_0 \leftarrow k_b$
$k_1 \leftarrow k_b + K_b - 1$
while $k_0 \le k_1$ do
    for $k \leftarrow$ from $k_0$ to $k_1$ do
        $n \leftarrow \mathrm{round}(k/h)$
        if $L_b(n) < Z_b(k)$ then
            $L_b(n) \leftarrow Z_b(k)$
    end
    $h \leftarrow h + 1$
    # harmonic $h+1$ is above the band
    $k_0 \leftarrow \mathrm{ceil}[(k_b + K_b)h/(h + 1)]$
    if $k_0 < k_b$ then $k_0 \leftarrow k_b$
    # harmonic $h-1$ is below the band
    $k_1 \leftarrow \mathrm{floor}[(k_b - 1)h/(h - 1)]$
    if $k_1 > k_b + K_b$ then $k_1 \leftarrow k_b + K_b$
end
```

The global F0 weights $L(n)$ and inharmonicity factors $\beta(n)$ do not need to be calculated for all fundamental frequency indices n. Instead, only a set of fundamental frequency indices $\{n_1, n_2, \ldots, n_Q\}$ is collected from the bandwise weight vectors $L_b(n)$. This is possible, and advantageous since if a sound is perceptible at all, it generally has a high weight in at least one of the bands. Selecting a couple of maxima from each band preserves the correct fundamental frequency among the candidates.

The maximum global weight $L(n)$ can be used as such to determine the true F0. However, an even more robust selec-

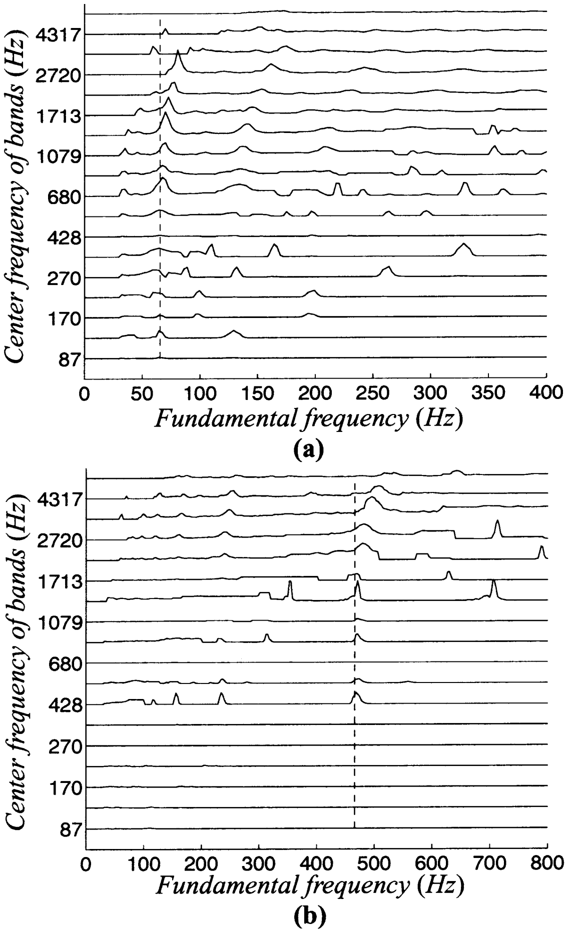

Fig. 4. Bandwise-calculated F0 weights $L_b(n)$ for two piano tones, Figure (a) with F0 65 Hz and Figure (b) with F0 470 Hz. The vectors are displaced vertically for clarity. The true pitches of the tones are indicated with dashed vertical lines.

tion among the candidates can be made by further inspecting the spectral smoothness of the F0s that have the highest global weights. This is the reason why a smoothing module is used in Fig. 1 before storing the F0. This module will be described in detail in Section III. For the sake of discussion in Section II-C one can assume that the maximum global score $L(n)$ determines the predominant F0.

C. Spectral Smoothness Principle

1) Iterative Estimation and Separation: The presented method is capable of making robust predominant-F0 detections in polyphonic signals. Moreover, the inharmonicity factor and precise frequencies of each harmonic partial of the detected sound are produced. A natural strategy for extending the presented algorithm to multiple-F0 estimation is to remove the partials of the detected sound from the mixture and to apply the predominant-F0 algorithm iteratively to the residual spectrum.

Detected sounds are separated in the frequency domain. Each sinusoidal partial of a sound is removed from the mixture spectrum in two stages. First, good estimates of the frequency and amplitude of the partials must be obtained. It is assumed that these parameters remain constant in the analysis frame. Second, using the found parameters, the spectrum in the vicinity of the partials is estimated and linearly subtracted from the mixture spectrum.

Initial estimates for the frequency and amplitude of each sinusoidal partial of a sound are produced by the predominant-F0 detection algorithm. Efficient techniques for estimating more precise values have been proposed e.g. in [42]. A method widely adopted is to apply Hamming windowing and zero padding in the time domain, to calculate Fourier spectrum, and to use quadratic interpolation of the spectrum around the partial. The second problem, estimating the spectrum in the vicinity of the partial is equivalent to translating the magnitude spectrum of the original analysis window at the frequency of the sinusoidal partial. For Hamming window without zero padding, it was found to be sufficient to perform the subtraction for five adjacent frequency bins.

2) The Problem of Coinciding Frequency Partials: One issue that is addressed in the algorithm is the problem of coinciding frequency partials. To illustrate this problem, simulations were run using the iterative procedure on randomly generated F0 mixtures. Fig. 5 shows the errors as a function of the musical intervals that occur in the erroneously transcribed sound mixtures (see Appendix). In most cases, the iterative approach works rather reliably. However, an important observation can be made when the distribution of the errors in Fig. 5 is analyzed. The error rate is strongly correlated with certain F0 relations. The conclusion to be noted is that a straightforward estimation and subtraction approach is likely to fail in cases where the fundamental frequencies of simultaneous sounds have simple rational number relations, also called *harmonic* relations. These are indicated over the corresponding bars in Fig. 5.

Coinciding frequency partials from different sounds can cause the algorithm to fail since many of the partials coincide in frequency. When the sound detected first is removed, the coinciding harmonics of remaining sounds are corrupted in the subtraction procedure. After several iterations, a remaining sound can become too corrupted to be correctly analyzed in the iterations that follow.

When two sinusoidal partials with amplitudes a_1 and a_2 and phase difference θ_Δ coincide in frequency, the amplitude of the resulting sinusoid can be calculated as

$$a_s = |a_1 + a_2 e^{i\theta_\Delta}|. \tag{11}$$

If the two amplitudes are roughly equivalent, the partials may either amplify or cancel each other, depending on θ_Δ. However, if one of the amplitudes is significantly greater than the other, as is usually the case, a_s approaches the maximum of the two.

Assuming ideal harmonicity, it is straightforward to prove that the harmonic partials of two sounds coincide if and only if the fundamental frequencies of the two sounds are in rational number relations. Moreover, if the harmonic indices of the coinciding partials are p and q, then every p^{th} partial of the first sound coincides with every q^{th} partial of the other sound. An important principle in Western music is to pay attention to the pitch relationships of simultaneously played notes. Simple harmonic relationships are favored over dissonant ones in order to make the sounds blend better. Because harmonic relationships

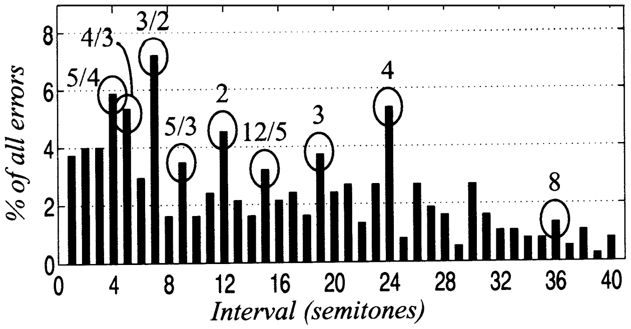

Fig. 5. Distribution of the F0 estimation errors as a function of the musical intervals that occur in the erroneously transcribed sound mixtures.

are so common in music, these “worst cases” must be handled well in general.

To solve this problem, the spectra of the detected sounds must be smoothed before subtracting them from the mixture. Consider the preprocessed spectrum $Z(k)$ of a two-sound mixture in Fig. 6(a). In the figure, the harmonic partials of the higher-pitched sound coincide with every third harmonic of the lower-pitched sound $(F_{higher} = 3F_{lower})$. As predicted by (11), the coinciding partials randomly cancel or amplify each other at the low frequencies, whereas at the higher frequencies the summary amplitudes approach the maximum of the two, i.e., the spectral envelope of the higher sound.

When the spectrum of the lower-pitched sound is smoothed (the thin slowly decreasing horizontal curve in Fig. 6(b)), the coinciding partials at the higher frequencies rise above the smooth spectrum and thus remain in the residual after subtraction. In particular, this solves a very common case where the dense harmonic series of a lower-pitched sound matches the few partials of a higher-pitched sound. Detecting the higher-pitched sound first is less common and in that case, only a minority of the harmonics of the lower-pitched sound are deleted.

It should be noted that simply smoothing the amplitude envelope (the thin curve in Fig. 6(b)) of a sound before subtracting it from the mixture does not result in lower error rates. A successful smoothing algorithm was found by applying psychoacoustic knowledge. The full motivation for this approach has been presented in [43] and is beyond the scope of this paper.

The algorithm first calculates a moving average over the amplitudes of the harmonic partials of a sound. An octave-wide triangular weighting window is centered at each harmonic partial h, and the weighted mean d_h of the amplitudes of the partials in the window is calculated. This is the smooth spectrum illustrated by a thin horizontal curve in Fig. 6(b). The original amplitude value a_h is then replaced with the minimum of the original (a_h) and d_h:

$$a_h \leftarrow min(a_h, d_h). \tag{12}$$

These values are illustrated by a thick curve in Fig. 6(b). Performing this straightforward smoothing operation before subtracting the sound from the mixture reduces the error rates significantly.

A further improvement to the smoothing method can be made by utilizing the statistical dependency of every p^{th} harmonic partial, as was previously explained following (11) in this section. The algorithm applies a multistage filter with the following steps [43]. First, the indices $\{\ldots, h-1, h, h+1, h+2, \ldots\}$ of the harmonic partials around harmonic h are collected from an octave-wide window. Next, the surrounding partials are classified into groups, where all the harmonics that share a common divisor are put in the same group, starting from the smallest prime factors. Third, weighted mean around harmonic h is calculated inside groups in the manner described above. In the last step, the estimates of different groups are averaged, weighting each group according to its mean distance from harmonic h.

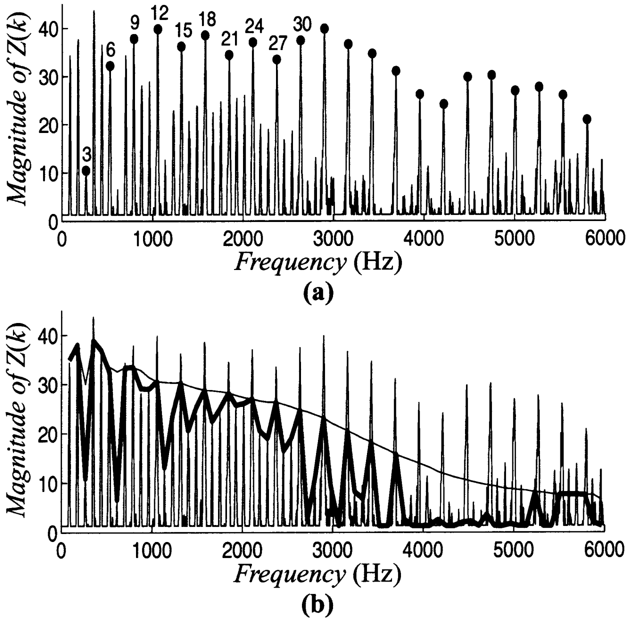

Fig. 6. Illustration of the spectral smoothness principle. (a) Preprocessed spectrum $Z(k)$ containing two sounds with F0s in the relation 1:3. (b) Two different smoothing operations have been used to estimate the spectral envelope of the lower-pitched sound. The results are indicated with thin and thick horizontal curves.

3) Recalculation of F0 Weights After Smoothing: The described principle of smoothing provides an efficient solution to another common class of errors. In this class of errors two or more fundamental frequencies in specific relationships may cause the detection of a nonexistent sound, such as the root of a musical chord in its absence. For instance, when two harmonic sounds with fundamental frequencies $2F$ and $3F$ are played, the spectra of these sounds match every second and every third harmonic partial of a nonexisting sound with fundamental frequency F. This frequency F may be erroneously estimated in the predominant-F0 calculations given the observed partials.

The problem can be solved by applying smoothing and an ordered search when selecting among the candidate indices n_i calculated by the predominant-F0 algorithm (see the end of Section II-B). First, the candidate n_1 with the highest global weight $L(n_1)$ is taken and its spectrum is smoothed. Then the weight of this candidate is recalculated using the smoothed harmonic amplitudes. In the above-described case of a nonexistent sound, the irregularity of the spectrum decreases the level of the smooth spectrum significantly, and the weight remains low. If the recalculated weight drops below the second-highest weight, the next candidate n_2 is processed, and this is continued. The highest recalculated global weight determines the F0. The computational load of applying smoothing and recalculation to select among the candidates is negligible, since the recalculation procedure has to consider only one F0 and one value of m in (7).

D. Estimating the Number of Concurrent Sounds

A mechanism is needed which controls the stopping of the iterative F0 estimation and sound separation process. This leads to the estimation of the number of concurrent sounds, i.e. the polyphony. The difficulty of the task is comparable to that of finding the F0 values themselves. Huron has studied musicians' ability to identify the number of concurrently sounding voices in polyphonic textures [44]. According to his report by four-voice polyphonies the test subjects underestimated the number of voices in more than half of the cases.

A statistical-experimental approach was taken to solve the problem. Random mixtures of one to six concurrent harmonic sounds were generated by allotting sounds from McGill University Master Samples collection [45]. The mixtures were then contaminated with pink noise or random drum sounds from Roland R-8 mk II drum machine. Signal-to-noise ratio was varied between 23 dB and -2 dB.

The behavior of the iterative multiple-F0 estimation system was investigated using these artificial mixtures with known polyphonies. Based on this investigation it was decided to split the estimation task into two stages. The first stage detects if there are any harmonic sounds at all in the input, and the second estimates the number of concurrent sounds, if the first test has indicated that some are present. It was found that the best single feature to indicate the presence of harmonic sounds was the global weight $L_{max}^{(1)}$ of the winning F0 candidate at the first iteration. The best compound feature consists of $L_{max}^{(1)}$ and terms related to the signal-to-noise ratio (SNR) of the input signal:

$$v_0 = 4\ln[L_{max}] + \ln\left[\sum_{l=k_0}^{k_1} X(l)\right] - \ln\left[\sum_{l=k_0}^{k_1} \hat{N}_{(\mathrm{pow})}(1)\right]. \tag{13}$$

Here $X(k)$ is the discrete power spectrum of the input signal and $\hat{N}_{(\mathrm{pow})}(k)$ is the power spectrum of the estimated noise, obtained by applying inverse transform of (2) on $\hat{N}(k)$. Frequency indices k_0 and k_1 are the same as in (3). A signal is determined to contain harmonic sounds when v_0 is greater than a fixed threshold.

If an analysis frame has been determined to contain harmonic sounds, another model is used to estimate the number of sounds. The maximum global weight $L_{max}^{(i)}$ at iteration i was again the best single feature for controlling the iteration stopping. However, the weight values are affected by the SNR $L_{max}^{(i)}$ getting smaller in noise. The bias can be explicitly corrected, resulting in the measure

$$v_i = 1.8\ln\left(L_{max}^{(i)}\right) - \ln\left[\sum_{l=k_0}^{k_1} X(l)\right] + \ln\left[\sum_{l=k_0}^{k_1} \hat{N}_{(\mathrm{pow})}(1)\right]. \tag{14}$$

As long as the value of v_i stays above a fixed threshold, the sound detected at iteration i is accepted as a valid F0 estimate and the iteration is continued. In (13) and (14), the SNR-related terms have different roles and thus different signs.

III. Results

A. Experimental Setup

Simulations were run to validate the proposed methods. The acoustic database consisted of samples from four different sources. The McGill University Master Samples collection [45] and independent recordings for acoustic guitar were available already during the development phase of the system. In order to verify that the results generalize outside these data sets, the samples from the University of Iowa website [46] and IRCAM Studio Online [47] were added to the final evaluation set. There were altogether 30 different musical instruments, comprising brass and reed instruments, strings, flutes, the piano, and the guitar. These introduce several different sound production mechanisms and a variety of spectra. On the average, there were 1.8 pieces of each of the 30 instruments and 2.5 different playing styles per instrument. The total number of samples was 2536. These were randomly mixed to generate test cases. The instruments marimba and the vibraphone were excluded from the data set since their spectrum is quite different from the others and extremely inharmonic. The system admittedly cannot handle these sounds reliably.

Semirandom sound mixtures were generated according to two different schemes. *Random mixtures* were generated by first allotting an instrument and then a random note from its whole playing range, restricting, however, the pitch over five octaves between 65 Hz and 2100 Hz. The desired number of simultaneous sounds were allotted and then mixed with equal mean-square levels. *Musical mixtures* were generated in a similar manner, but favoring different pitch relationships according to a statistical profile discovered by Krumhansl in classical Western music [48, p. 68]. In brief, octave relationships are the most frequent, followed by consonant musical intervals, and the smallest probability of occurrence is given to dissonant intervals. In general, musical mixtures are more difficult to resolve (see Section II-C2).

Acoustic input was fed to the multiple-F0 algorithm that estimated F0s in a single time frame. Unless otherwise stated, the number of F0s to extract, i.e., the polyphony, was given along with the mixture signal. It was found to be more informative to first evaluate the multiple-F0 estimator without the polyphony estimator, because these two are separable tasks and because the reference methods do not implement polyphony estimation. The configuration and parameters of the system were fixed unless otherwise stated. A correct F0 estimate was defined to deviate less than half a semitone (±3%) from the true value, making it "round" to a correct note on a Western musical scale. Errors smaller than this are not significant from the point of view of music transcription.

B. Reference Methods

To put the results in perspective, two reference methods were used as a baseline in simulations. The first method, *YIN*, is a state-of-the-art *monophonic* F0 estimator for speech and music signals [49]. Naturally, the method can be used as a baseline in single-F0 analysis only. The algorithm has been designed to be reliable for individual analysis frames and has been thoroughly tested and compared with other methods in [49]. The original implementation by the authors was employed and parameters were left intact except the "absolute threshold" which was fine-tuned to value 0.15 to improve the performance.

The other reference method, referred to as *TK*, is a multiple-F0 estimator proposed by Tolonen and Karjalainen in [21]. The implementation was carefully prepared based on the reference, and the original code by the authors was used in the warped linear prediction part of the algorithm. Thorough testing was carried out to verify the implementation. Original parameters given in [21] were applied. As reported by the authors, the method cannot handle "spectral pitch," i.e., F0s above 1 kHz. It was further found out here that the method is best at detecting F0s in the three-octave range between 65 Hz and 520 Hz. Thus, in the simulations that follow, the mixtures given to the *TK* method were restricted to contain F0s below either 520 Hz or 1 kHz. The bound is specified for each case in the simulation results to follow.

C. Experimental Results

In the first experiment, different F0 estimators are compared. For this experiment, a predominant-F0 estimate (firstly detected F0) was defined to be correct if it matches the correct F0 of *any* of the component sounds. That is, only a single match among all possible F0s is required in this error measure. The error rate was calculated as the amount of predominant-F0 errors divided by the number of random sound mixtures (1000), not by the number of reference notes (e.g. 6000 in the six-note mixtures). F0 estimation was performed in a single 190 ms time frame 100 ms after the onset of the sounds. Fig. 7 shows the error rates for the predominant-F0 estimation in different polyphonies. Results are given for the proposed system and for the two reference systems.

For the proposed system, the error rates are generally below 10%, getting close only in six-note polyphonies. Surprisingly, increasing the number of concurrent sounds from one to two appears to help lower the error rate of detecting at least one F0 correctly. However, this is due to the fact that the acoustic database contains a small percentage of irregular sounds for which the simple model in (7) does not work. Among these are e.g. high flute tones and high plucked string tones. Two-sound mixtures are more likely to contain at least one clear sound with no anomalities, which then appears as the predominant F0.

The *YIN* method achieves 4.1% error rate for isolated notes. Since the method is not intended for multiple-F0 estimation, it is not fair to make comparison for polyphonic signals. Like other single-F0 estimators, the algorithm converges to 70% error rate already in three-note mixtures. The *TK* method is not quite as reliable for single-pitch signals, but works robustly in polyphony. If the method is given F0s only below 520 Hz, the predominant-F0 detection accuracy comes close to the proposed system in higher polyphonies. This is partly due to the relatively higher random guess rate.

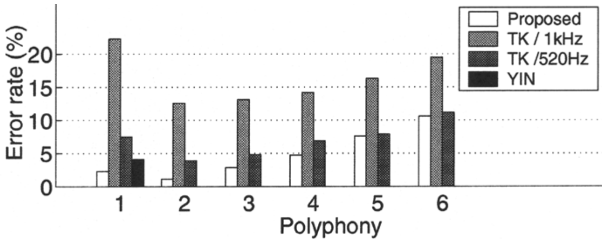

Fig. 7. Error rates for detecting any of the potential F0s in a sound as a function of the predominant-F0 estimation algorithm and the polyphony.

In the second experiment, the performance of multiple-F0 estimation is explored in more detail. For multiple-F0 estimation, a more appropriate error measure is a *note error rate (NER)* metric. The NER is defined as the sum of the F0s in error divided by the number of F0s in the reference transcription. The errors are of three types:

a) Substitution errors. These are defined as errors in which a given F0 is detected but the estimated value differs more than ±3% from the reference.

b) Deletion errors have occurred if the number of detected F0s is smaller than the number of F0s in the reference.

c) Insertion errors have occurred if the number of detected F0s exceeds that in the reference.

Substitution and deletion errors together can be counted from the number of F0s in the reference that are not correctly estimated. Insertion errors can be counted from the number of excessive estimates.

Results for multiple-F0 estimation in different polyphonies are shown in Fig. 8. Here the number of concurrent sounds to extract was given for each mixture signal, i.e., the polyphony was known. Thus insertion and deletion errors do not occur. Random and musical sound mixtures were generated according to the described schemes, and the estimator was then requested to find a given number of F0s in a single 190 ms time frame 100 ms after the onset of the sounds.

In Fig. 8, the bars represent the overall NER's as a function of the polyphony. As can be seen, the NER for random four-sound polyphonies is 9.9% on the average. The different shades of gray in each bar indicate the error cumulation in the iteration, errors which occurred in the first iteration at the bottom, and errors of the last iteration at the top. As a general impression, the system works reliably and exhibits graceful degradation in increasing polyphony. Results for musical mixtures are slightly worse than for random mixtures (see Section II-C2), but the difference is not great. This indicates that the spectral smoothing principle works well in resolving harmonically related pitch combinations.

Analysis of the error cumulation reveals that the errors which occurred in the last iteration account for approximately half of the errors in all polyphonies, and the probability of error increases rapidly in the course of iteration. Besides indicating that the subtraction process does not work perfectly, the conducted listening tests suggest that this is a feature of the problem itself, rather than only a symptom of the algorithms used. In most mixtures, there is a sound or two that are very difficult to perceive because their spectrum is virtually hidden under the other sounds.

For the reference method *TK*, note error rates for mixtures ranging from one to six sounds were 22%, 31%, 39%, 45%,

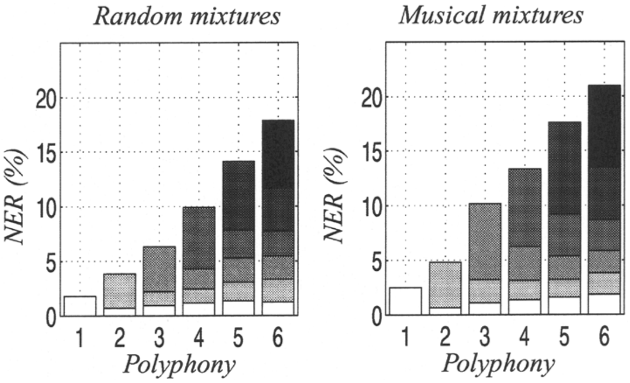

Fig. 8. Note error rates for multiple-F0 estimation using the proposed algorithm when the polyphony was known. Bars represent the overall error rates, and the different shades of gray the error cumulation in iteration.

49%, and 53%, respectively, when F0s were resticted to range 65 Hz–1 kHz. For the three-octave range between 65 Hz and 520 Hz, the corresponding error rates were 7.5%, 17%, 26%, 34%, 38%, and 43%. Given the complexity of the problem, even these error rates are rather low.

Table II gives the error rates for different system configurations. Different processing elements were disabled one-by-one in order to evaluate their importance. In each case, the system was kept otherwise fixed. In the first test, the mechanisms that accommodate inharmonicity were disabled. One mechanism is in bandwise F0-weight calculations, and in this case the offset m in (7) was constrained to a value which corresponds to an ideal harmonicity. Another mechanism is in the integration phase. Here the inharmonicity factor was constrained to zero, leading to a straightforward summing across squared weight vectors. The resulting performance degradation is mostly due to the bandwise calculations.

In the second test, the spectral smoothing algorithm was switched between the one presented in Section II-C2 and a version which leaves the harmonic series intact. The smoothing operation made a significant improvement to multiple-F0 estimation accuracy in all polyphonies, except for the single-note case where it did not have a noticeable effect on the performance.

In all the results presented above, the polyphony of the signals was known. Fig. 9 shows the statistical error rate of the overall multiple-F0 estimation system when the polyphony is estimated in the analysis frame, as described in Section II-D. Results are shown for two different polyphony estimation thresholds (i.e., thresholds for v_i in (14) which were 0.65 and 1.1 for the left and right panels, respectively). Depending on the application, either overestimating or underestimating the number of concurrent sounds may be more harmful. In a music transcription system, for example, extraneous notes in the output are very disturbing. However, if the frame-level F0 estimates are further processed at a higher level, it is usually advantageous to produce too many rather than too few note candidates.

In general, the proposed polyphony estimation method operates robustly. However, when the estimation threshold is tuned to avoid extraneous detections in monophonic signals, the polyphony is underestimated in higher polyphonies. On the other hand, when underestimations are avoided, many of the extraneous F0s appear in monophonic signals. This is likely to be characteristic of the problem itself (see Huron's report [44] mentioned in Section II-D). One or two sounds in rich polyphonies are usually very difficult to distinguish.

TABLE II
ERROR RATES FOR DIFFERENT SYSTEM CONFIGURATIONS WHEN THE POLYPHONY OF THE SIGNALS WAS KNOWN

System configuration	Polyphony	
	1	4
Complete system	1.8 %	9.9 %
Inharmonicity not allowed	6.2 %	17 %
No smoothing	2.2 %	20 %

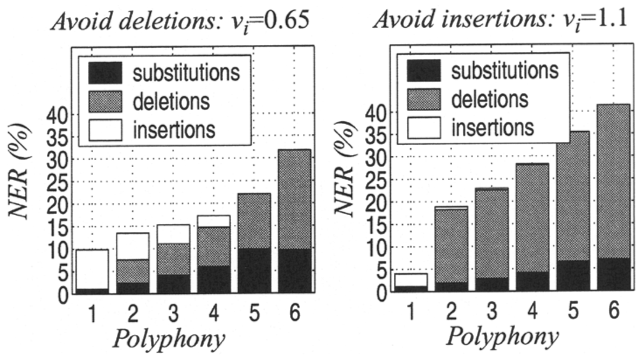

Fig. 9. Error rates for the two different polyphony estimations strategies.

Table III shows the influence of shortening the analysis frame. The significant difference between 190 ms and 93 ms frame sizes is partly caused by the fact that the applied technique was sometimes not able to resolve the F0 with the required ±3% accuracy. Also, irregularities in the sounds themselves, such as vibrato, are more difficult to handle in short frames. However, when the time frame was shortened from 190 ms to 93 ms, the error rate of the reference method *TK* increased only by approximately 5% for both 1000 Hz and 520 Hz F0 limits and in all polyphonies. Thus, the error rates of *TK* were essentially the same as those presented around Fig. 8. While the performance is still clearly worse than that of the proposed method (polyphony was known), an obvious drawback of the proposed method is that its accuracy depends on the length of the analysis frame. A basic reason for this is that the linear frequency resolution of spectral methods does not suffice at the low end, whereas the frequency resolution of autocorrelation-based methods is proportional to the inverse of frequency, being closer to the logarithmic frequency resolution of musical scales and human hearing. Despite these differences, reliable multiple-F0 estimation in general seems to require longer time frames than single-F0 estimation.

Fig. 10 shows the NER's in different types and levels of additive noise when the polyphony was known. Pink noise was generated in the band between 50 Hz and 10 kHz. Percussion instrument interference was generated by randomizing drum samples from a Roland R-8 mk II drum machine. The test set comprised 33 bass drum, 41 snare, 17 hi-hat, and 10 cymbal

TABLE III
ERROR RATES FOR DIFFERENT ANALYSIS FRAME LENGTHS

Polyphony estimation	Frame size	Actual polyphony					
		1	2	3	4	5	6
Polyphony known	190 ms	1.8	3.9	6.3	9.9	14	18
	93 ms	4.2	8.7	16	22	29	34
Estimate, avoid deletions	190 ms	11	14	16	18	22	32
	93 ms	14	19	23	30	38	46

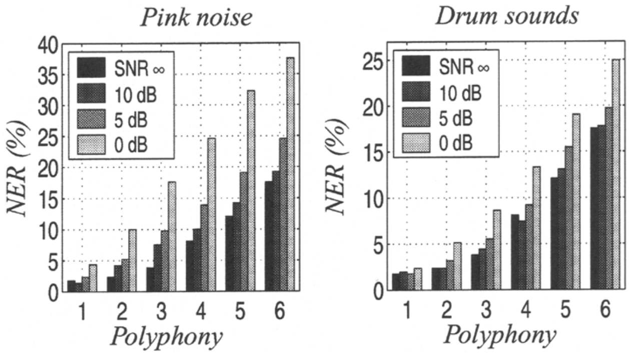

Fig. 10. Error rates in additive pink noise (left panel) and with interfering percussive sounds (right panel). For both noise types, error rates for a clean signal and for noisy signals with SNR's 10 dB, 5 dB, and 0 dB are given. Polyphony was known.

sounds. The signal-to-noise ratio was adjusted within the analysis frame, and the ratio was defined between the noise and the *sum* of the harmonic sounds. Thus, the SNR from the point of view of individual sounds is much worse in higher polyphonies. A 190 ms frame was applied.

D. Comparison With Human Performance

Listening tests were conducted to measure the human pitch identification ability, particularly the ability of trained musicians to transcribe polyphonic sound mixtures. Detailed analysis of the results is beyond the scope of this article. Only a summary of the main findings can be reviewed here.

Test stimuli consisted of computer-generated mixtures of simultaneously onsetting sounds that were reproduced using sampled Steinway grand piano sounds from the McGill University Master Samples collection [45]. The number of co-occurring sounds varied from two to five. The interval between the highest and the lowest pitch in each individual mixture was never wider than 16 semitones in order to make the task feasible for the subjects that did not have "absolute pitch", i.e., the rare ability of being able to name the pitch of a sound without a reference tone. Mixtures were generated from three pitch ranges (i.e., registers): low (33 Hz–130 Hz), middle (130 Hz–520 Hz), and high (520 Hz–2100 Hz). In total, the test comprised 200 stimuli.

The task was to write down the musical intervals, i.e., pitch relations, of the presented sound mixtures. Absolute pitch values were not asked for and the number of sounds in each mixture was given. Thus, the test resembles the musical interval and chord identification tests that are a part of the basic musical training in Western countries.

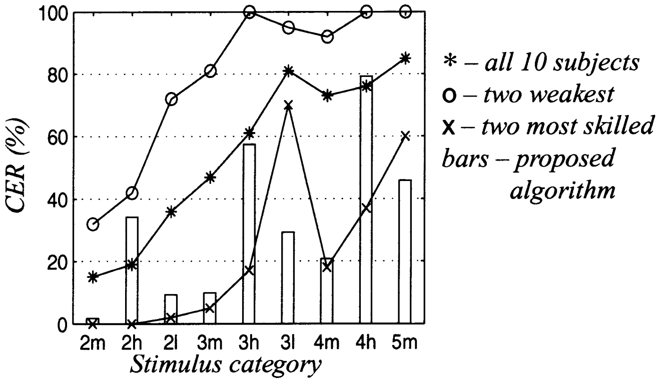

Fig. 11. Chord error rates of the human listeners (curves) and of the proposed algorithm (bars) for different stimulus categories. The lowest curve represents the two most skilled subjects, the middle curve the average of all subjects, and the highest curve the two clearly weakest subjects. The labels of the simulus categories consist of a number which signifies the polyphony, and of a letter which tells the pitch range used.

A total of ten subjects participated in the test. All of them were trained musicians in the sense of having taken several years of ear training[2] in music. Seven subjects were students of musicology at university level. Two were more advanced musicians, possessing absolute pitch and exceptional pitch identification abilities. One subject was an amateur musician of similar musical ability as the seven students.

Fig. 11 shows the results of the listening test. Chord error rates (CER) are plotted for different stimulus categories. CER is the percentage of sound mixtures where one or more pitch identification errors occurred. The labels of the categories consist of a number which signifies the polyphony, and of a letter which tells the pitch range used. Letter "m" refers to the middle, "h" to the high, and "l" to the low register. Performance curves are averaged over three different groups. The lowest curve represents the two most skilled subjects, the middle curve the average of all subjects, and the highest curve the two clearly weakest subjects.

The CER's cannot be directly compared to the NER's given in Fig. 8. The CER metric is more demanding, accepting only sound mixtures where all pitches are correctly identified. It had to be adopted to unambiguously process the musicians' answers, which were given as pitch intervals.

For the sake of comparison, the stimuli and performance criteria used in the listening test were used to evaluate the proposed computational model. Five hundred instances were generated from each category included in Fig. 11, using the same software that randomized samples for the listening test. These were fed to the described multiple-F0 system. The CER metric was used as a performance measure.

The results are illustrated with bars in Fig. 11. As a general impression, only the two most skilled subjects perform better than the computational model. However, performance differences in high and low registers are quite revealing. The devised algorithm is able to resolve combinations of low sounds that are beyond the ability of human listeners. This seems to be due to the good frequency resolution applied. On the other hand,

[2] The aim of ear training in music is to develop the faculty of discriminating sounds, recognizing musical intervals, and playing music by ear, i.e., without the aid of written music.

human listeners perform relatively well in the high register. This is likely to be due to an efficient use of the temporal features, onset asynchrony and different decay rates, of high piano tones. These were not available in the single time frame given to the multiple-F0 algorithm.

IV. Conclusions

The paper shows that multiple-F0 estimation can be performed reasonably well using only spectral cues, harmonicity and spectral smoothness, without the need for additional long-term temporal features. For a variety of musical sounds, a prior knowledge of the type of sound sources involved is not necessary, although adaptation of internal source (e.g. instrument) models would presumably further enhance the performance.

The primary problem in multiple-F0 estimation appears to be in associating partials correctly with their individual sources of production. The harmonicity principle must be applied in a manner that is flexible enough to accommodate a realistic amount of inharmonicity in sound production, and yet constraining enough to prevent erroneous groupings. Contrasted with the complexity needed in handling inharmonicity, the harmonic summation model used to calculating F0 weights from the amplitudes of the grouped partials is very simple, as embodied in (8) and (9).

A spectral smoothing approach was proposed as an efficient new mechanism in multiple-F0 estimation and spectral organization. The introduction of this principle corrected approximately half of the errors occurring in a system which was otherwise identical but did not use the smoothness principle.

An attractive property of the iterative estimation and separation approach is that at least a couple of the most prominent F0s can be detected even in very rich polyphonies. The probability of error increases rapidly in the course of the iteration, but on the basis of the listening tests it was suggested that this is at least in part due to the inherent characteristics of the problem itself. The last iteration, i.e., estimation of the F0 of the sound detected last, accounts for approximately half of the errors in all polyphonies.

The main drawback of the presented method is that it requires a relatively long analysis frame in order to operate reliably for low-pitched sounds. This is largely due to the fact that the processing takes place in the frequency domain where sufficiently fine frequency resolution is required for harmonic series of low-pitched sounds.

The described method has been applied to the automatic transcription of continuous music on CD recordings. Some demonstration signals are provided at [50]. Contrary to the musical chord identification task, however, the accuracy is not comparable to that of trained musicians. There are several possibilities that can be explored as areas of future development. Integration across multiple time frames can be used to improve performance. While independent multiple-F0 estimation in each time frame is important for feature extraction, it does not account for the real experience represented in a human listener. Analogous to the case of speech recognition in which models of words and language are used to improve performance, use of higher-level features in music are also expected to improve music estimation and transcription tasks.

TABLE IV
Some Basic Musical Intervals

Interval name	Size (semitones)	F0 relation
octave	12	2:1
perfect fifth	7	3:2
perfect fourth	5	4:3
major third	4	5:4
minor third	3	6:5
major second	2	9:8

Appendix

Western music typically uses a *well-tempered* musical scale. That is, the notes are arranged on a logarithmic scale where the fundamental frequency F_k of a note k is $F_k = 440 \times 2^{(k/12)}$ Hz.

The notes on a standard piano keyboard range from $k = -48$ up to $k = 39$. The term *semitone* refers to the interval between two adjacent notes and is used to measure other musical intervals. The F0 relation of two notes that are one semitone apart is $F_{k+1}/F_k = 2^{(1/12)} \approx 1.06$.

Although the well-tempered scale is logarithmic, it can surprisingly accurately generate F0s that are in rational number relations. Table IV lists some basic musical intervals and the corresponding ideal rational number relations. Intervals which approximate simple rational number relationships are called *harmonic*, or, *consonant* intervals, as opposed to *dissonant* intervals.

References

[1] S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press, 1990.

[2] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal, “A comparative performance study of several pitch detection algorithms,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, no. 5, pp. 399–418, 1976.

[3] W. J. Hess, “Pitch and voicing determination,” *Advances in Speech Signal Processing*, 1991.

[4] A. de Cheveigné and H. Kawahara, “Comparative evaluation of F0 estimation algorithms,” in *Proc. Eurospeech*, Copenhagen, Denmark, 2001, pp. 2451–2454.

[5] W. M. Hartmann, “Pitch, periodicity, and auditory organization,” *J. Acoust. Soc. Amer.*, vol. 100, no. 6, pp. 3491–3502, 1996.

[6] J. A. Moorer, “On the transcription of musical sound by computer,” *Comput. Music J.*, pp. 32–38, Nov. 1977.

[7] C. Chafe and D. Jaffe, “Source separation and note identification in polyphonic music,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, 1986, pp. 1289–1292.

[8] R. C. Maher, “Evaluation of a method for separating digitized duet signals,” *J. Audio Eng. Soc.*, vol. 38, no. 12, pp. 956–979, 1990.

[9] H. Katayose and S. Inokuchi, “The Kansei music system,” *Comput. Music J.*, vol. 13, no. 4, pp. 72–77, 1989.

[10] M. Hawley, “Structure Out of Sound,” Ph.D. dissertation, MIT Media Laboratory, Cambridge, MA, 1993.

[11] L. Rossi, “Identification de sons polyphoniques de piano,” Ph.D. thesis, L’Universite de Corse, Corsica, France, 1998.

[12] D. F. Rosenthal and H. G. Okuno, Eds., *Computational Auditory Scene Analysis*. Mahwah, NJ: Lawrence Erlbaum, 1998.

[13] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, “Organization of hierarchical perceptual sounds: music scene analysis with autonomous processing modules and a quantitative information integration mechanism,” in *Proc. Int. Joint Conf. Artificial Intelligence*, Montréal, QC, Canada, 1995.

[14] K. D. Martin, “Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing,” Massachusetts Institute of Technology Media Laboratory Perceptual Computing Section, Tech. Rep. 399, 1996.

[15] D. P. W. Ellis, “Prediction-Driven Computational Auditory Scene Analysis,” Ph.D. thesis, MIT Media Laboratory, Cambridge, Massachusetts, 1996.

[16] M. Goto, “A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings,” in *Proc. IEEE International Conf. on Acoust., Speech, and Signal Processing*, Istanbul, Turkey, June 2000.

[17] G. J. Brown and M. P. Cooke, “Perceptual grouping of musical sounds: a computational model,” *J. New Music Res.*, vol. 23, pp. 107–132, 1994.

[18] D. Godsmark and G. J. Brown, “A blackboard architecture for computational auditory scene analysis,” *Speech Commun.*, vol. 27, pp. 351–366, 1999.

[19] R. Meddis and M. J. Hewitt, “Virtual pitch and phase sensitivity of a computer model of the auditory periphery I: pitch identification,” *J. Acoust. Soc. Amer.*, vol. 89, no. 6, pp. 2866–2882, 1991.

[20] R. Meddis and L. O’Mard, “A unitary model of pitch perception,” *J. Acoust. Soc. Amer.*, vol. 102, no. 3, pp. 1811–1820, 1997.

[21] T. Tolonen and M. Karjalainen, “A computationally efficient multipitch analysis model,” *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 708–716, Nov. 2000.

[22] A. de Cheveigné and H. Kawahara, “Multiple period estimation and pitch perception model,” *Speech Commun.*, vol. 27, pp. 175–185, 1999.

[23] R. Meddis and M. J. Hewitt, “Modeling the identification of concurrent vowels with different fundamental frequencies,” *J. Acoust. Soc. Amer.*, vol. 91, no. 1, pp. 233–245, 1992.

[24] A. de Cheveigné, “Separation of concurrent harmonic sounds: fundamental frequency estimation and a time-domain cancellation model of auditory processing,” *J. Acoust. Soc. Amer.*, vol. 93, no. 6, pp. 3271–3290, 1993.

[25] W. A. Sethares and T. W. Staley, “Periodicity transforms,” *IEEE Trans. Signal Processing*, vol. 47, no. 11, pp. 2953–2964, 1999.

[26] T. Nakatani and H. G. Okuno, “Harmonic sound stream segregation using localization and its application to speech stream segregation,” *Speech Commun.*, vol. 27, pp. 209–222, 1999.

[27] T. W. Parsons, “Separation of speech from interfering speech by means of harmonic selection,” *J. Acoust. Soc. Amer.*, vol. 60, pp. 911–918, 1976.

[28] T. Verma, “A Perceptually Based Audio Signal Model With Application to Scalable Audio Compression,” Ph.D. dissertation, Stanford University, 2000.

[29] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[30] H. Hermansky, N. Morgan, and H.-G. Hirsch, “Recognition of speech in additive and convolutive noise based on RASTA spectral processing,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Minneapolis, Minnesota, 1993.

[31] A. P. Klapuri, “Automatic transcription of musical recordings,” in *Proc. Consistent and Reliable Acoustic Cues Workshop*, Aalborg, Denmark, Sep. 2001.

[32] D. Talkin, “A robust algorithm for ptch tracking,” in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds: Elseview Science B.V., 1995.

[33] J. C. Brown and B. Zhang, “Musical frequency tracking using the methods of conventional and “narrowed” autocorrelation,” *J. Acoust. Soc. Amer.*, vol. 89, no. 5, pp. 2346–2354, 1991.

[34] A. P. Klapuri, “Qualitative and quantitative aspects in the design of periodicity estimation algorithms,” in *Proc. European Signal Processing Conference*, Tampere, Finland, Sept. 2000.

[35] B. Doval and X. Rodet, “Estimation of fundamental frequency of musical sound signals,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, 1991, pp. 3657–3660.

[36] J. C. Brown, “Musical fundamental frequency tracking using a pattern recognition method,” *J. Acoust. Soc. Amer.*, vol. 92, no. 3, pp. 1394–1402, 1992.

[37] M. Lahat, R. J. Niederjohn, and D. A. Krubsack, “A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 6, pp. 741–750, 1987.

[38] N. Kunieda, T. Shimamura, and J. Suzuki, “Robust method of measurement of fundamental frequency by ACLOS—autocorrelation of log spectrum,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, 1996, pp. 232–235.

[39] A. J. M. Houtsma, “Pitch perception,” in *Hearing—Handbook of Perception and Cognition*, B. J. C. Moore, Ed. San Diego, CA: Academic, 1995.

[40] N. F. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, 2nd ed. New York: Springer-Verlag, 1998.

[41] A. P. Klapuri, “Wide-band pitch estimation for natural sound sources with in harmonicities,” in *Proc. 106th Audio Eng. Soc. Convention*, Munich, Germany, 1999.

[42] X. Rodet, “Musical sound signal analysis/synthesis: sinusoidal + residual and elementary waveform models,” in *Proc. IEEE Time-Frequency and Time-Scale Workshop*, Coventry, U.K., Aug. 1997.

[43] A. P. Klapuri, “Multipitch estimation and sound separation by the spectral smoothness principle,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, 2001, pp. 3381–3384.

[44] D. Huron, “Voice denumerability in polyphonic music of homogeneous timbres,” *Music Perception*, vol. 6, no. 4, pp. 361–382, Summer 1989.

[45] F. Opolko and J. Wapnick, *McGill University Master Samples* . Montreal, QC, Canada: McGill University, 1987.

[46] The University of Iowa Musical Instrument Samples*http://theremin.music.uiowa.edu/* [Online]

[47] IRCAM Studio Online*http://soleil.ircam.fr/* [Online]

[48] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*. New York: Oxford Univ. Press, 1990.

[49] A. de Cheveigné and H. Kawahara, “YIN, a fundamental frequency estimator for speech and music,” *J. Acoust. Soc. Amer.*, vol. 111, no. 4, April 2002.

[50] Automatic Transcription of Music Demonstrations, A. P. Klapuri.*http://www.cs.tut.fil/~klap/iiro/* [Online]

Anssi P. Klapuri was born in Kälviä, Finland, in 1973. He received the M.Sc. degree in information technology from the Tampere University of Technology (TUT), Tampere, Finland, in June 1998. He is currently pursuing a postgraduate degree.

He has been with the TUT Institute of Signal Processing since 1996. His research interests include automatic transcription of music, audio content analysis, and signal processing.