A Parametric Method for Pitch Estimation of Piano Tones
Valentin Emiya, Bertrand David, Roland Badeau

To cite this version:

HAL Id: inria-00544147
https://inria.hal.science/inria-00544147
Submitted on 7 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A PARAMETRIC METHOD FOR PITCH ESTIMATION OF PIANO TONES

Valentin EMIYA, Bertrand DAVID, Roland BADEAU

Ecole Nationale Supérieure des Télécommunications - Département TSI
46 rue Barrault, 75634 PARIS cedex 13 - France

ABSTRACT
The efficiency of most pitch estimation methods declines when the analyzed frame is shortened and/or when a wide fundamental frequency ($F_0$) range is targeted. The technique proposed herein jointly uses a periodicity analysis and a spectral matching process to improve the $F_0$ estimation performance in such an adverse context: a 60ms-long data frame together with the whole, $\frac{T_4}{4}$-octaves, piano tessitura. The enhancements are obtained thanks to a parametric approach which, among other things, models the inharmonicity of piano tones. The performance of the algorithm is assessed, is compared to the results obtained from other estimators and is discussed in order to characterize their behavior and typical misestimations.

Index Terms— audio processing, pitch estimation

1. INTRODUCTION
Numerous methods dedicated to fundamental frequency ($F_0$) estimation of periodic signals try to extract the signal self-similarities by maximizing a function of time or frequency. In this manner, they measure a degree of internal resemblance in the waveform (ACF [1, 2], AMDF [3, 4], cepstrum [5]) or in the spectrum [6]. When processing real world musical sounds, these techniques are confronted to deviations from the theoretical model, such as the presence of noise, which can be both stationary and non stationary, or the possibly non-uniform distribution of the harmonics.

The development and applications of the quoted methods often deal with an extension to subband processing [2, 7], to an optimization of the main function [4, 7] or to the joint use of both time and frequency domains [8]. Typical errors that usually occur give a general idea of the difficulties the $F_0$ estimation task must cope with. Temporal or spectral methods tend to make sub-octave or octave errors respectively. Both of them come up against difficulties like a possibly non-uniform distribution of the harmonics.

2. PITCH ESTIMATION SYSTEM
2.1. High Resolution analysis
The $N_a$-length analyzed waveform is modeled by:

$$s(t) = \sum_{k=1}^{K} \alpha_k z_k^a + w(t)$$

defined for $t \in [0, N_a - 1]$ and composed of a sum of $K$ exponentially-modulated sinusoids $\alpha_k z_k^a$, $k \in [1, K]$ with complex amplitudes $\alpha_k = A_k e^{i\phi_k} \in \mathbb{C}$, $(A_k$ being the real, positive amplitude and $\phi_k$ the initial phase), and distinct poles $z_k = e^{i\phi_k} + \frac{1}{d_k}$ (where $f_k$ being the frequency and $d_k$ the damping factor) plus an additive colored noise $w(t)$. This section details how the signal is preprocessed, how poles $z_k$ are then estimated via the ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) algorithm [10], and how amplitudes $\alpha_k$ are finally extracted.

Preprocessing. A two-step preprocessing stage is applied to the signal sampled at 32 kHz:

1. The cubic computational cost of the ESPRIT algorithm is reduced when the number of poles to be estimated is low. This is achieved by using a filter bank. The signal is split into $D = 32$ subbands with width 500 Hz by using cosine-modulated filters [11]. The order of magnitude of the computational cost drops from $N_a^2$ to $N_a^2/D^2$ ($N_a^2/D^3$ per band) leading to a satisfactory processing time for the analysis bloc.

2. Components of piano sounds are particularly well represented by the exponential sinusoidal plus noise model introduced in (1). However, the ESPRIT algorithm only applies to the restrictive case of white noise. Thus, the second preprocessing step consists in whitening the noise in each subband thanks to an AR filter estimated on the smoothed spectrum of the signal.

ESPRIT algorithm. The signal in each preprocessed subband is a sum of exponentially-modulated sinusoids plus white noise. Assuming the number of poles is known, the ESPRIT algorithm [10]
gives an estimation of those poles. The method is based on a subspace projection on the so-called signal subspace and benefits from the rotational invariance property of this signal subspace.

**Estimation of the number of poles.** In the current application, the number of poles in each subband is not known a priori and thus must be estimated. The ESTER [12] algorithm establishes a criterion \( J(p) \) that provides an estimation of the number of poles as \( \arg\max_{p \in \mathbb{P}} (J(p) > \delta_j) \), \( P \) being the set of candidates for the number of poles and \( \delta_j \) a threshold tuned to \( \delta_j = 10 \) in the current study. The result obtained by this method is either correctly estimated, or slightly over-estimated. As shown in [12], the latter case is not disturbing for the ESPRIT analysis, and weak amplitudes are estimated for the spurious poles.

**Estimation of amplitudes.** Once the poles extracted, amplitudes are estimated by a least squares algorithm applied to the subband signal. The effects of the preprocessing stage on the amplitudes in each subband are corrected by applying the inverse filters of the various preprocessing steps – whitening, filter bank and preemphasis filter series –, leading to the estimation of the amplitudes \( \alpha_k, k \in [1, K] \).

### 2.2. Pitch estimation

#### 2.2.1. Temporal method

Periodicity is often analyzed by assuming the signal is an observation of a real, wide-sense stationary (WSS) process \( y \) and uniformly distributed along \([0, \pi] \). The effects of the preprocessing stage on the amplitudes in each subband are corrected by applying the inverse filters of the various preprocessing steps – whitening, filter bank and preemphasis filter series –, leading to the estimation of the amplitudes \( \alpha_k, k \in [1, K] \).

A parametric amplitude spectrum is designed from the estimates of amplitudes \( f_k \) and energies \( E_k \) of components \( k \in [1, K] \). It is composed of a sum of \( K \) gaussian curves centered in \( 2\pi f_k \alpha_k \), initial phases \( \phi_k \), which are assumed to be independent

#### 2.2.2. Spectral method

A parametric amplitude spectrum is designed from the estimates of frequencies \( f_k \) and energies \( E_k \) of components \( k \in [1, K] \). It is composed of a sum of \( K \) gaussian curves centered in \( 2\pi f_k \alpha_k \), initial phases \( \phi_k \), which are assumed to be independent and uniformly distributed along \([0, 2\pi]\). The autocovariance function of \( y \) is \( R_y(\tau) = \sum_{\tau=1}^{K} 2\alpha_k^2 \cos(2\pi f_k \tau/2) + \delta f_k^2 \). Therefore we can define a temporal function \( R(\tau) \) for \( F_0 \) estimation from the parameters estimated by the high resolution analysis:

\[
R(\tau) = \sum_{k=1}^{K} p_k \cos(2\pi f_k \tau)
\]

\[
p_k = \left\{ \begin{array}{ll}
|\alpha_k|^2 & \text{if } |z_k| = 1 \\
|\alpha_k|^2 (1 - |z_k|^2) & \text{otherwise}
\end{array} \right.
\]

where \( \tau > 0 \), \( f_k = \frac{\omega h}{2\pi} \) is the normalized frequency of component \( k \), and the instantaneous power \( p_k \) is an estimate of coefficient \( 2\alpha_k^2 \) over the analysis frame.

In the case of a slightly inharmonic sound, the frequency deviation weakens or even removes the maxima of \( R(\tau) \) at the multiples of the period. The inharmonicity law [13] for a piano tone of fundamental frequency \( f_0 \) causes partials \( h \) not to be located at frequency \( h_f \) but at \( h_f \sqrt{1 + \beta (h^2 - 1)} \), \( \beta \) being the inharmonicity coefficient of the note. As illustrated in fig. [1], this frequency stretching may be inversed by remapping the set of estimated frequencies \( \{f_k, k \in [1, K]\} \) to a set of frequencies \( \{g_{f_0,k}, k \in [1, K]\} \):

\[
g_{f_0,k} = \frac{f_k}{\sqrt{1 + \beta (f_0) (h^2 (f_0, f_k) - 1)}}
\]

As the remapping process causes the remapped frequencies \( g_{f_0,k} \) of the partials to be perfect multiples of the actual fundamental frequency \( f_0 \), we replace \( f_k \) with \( g_{f_0,k} \) in (2) to obtain a temporal function \( R_{inh}(\tau) \) for piano tones which is maximum for \( \tau = \frac{1}{\tau_0} \):

\[
R_{inh}(\tau) = \sum_{k=1}^{K} p_k \cos \left( 2\pi g_{f_0,k} \tau \right)
\]

\[
\sigma = \text{set to } \tau_{min}/4 \text{ where } \tau_{min} \text{ is the lower bound of the } F_0 \text{ search range in order to prevent overlap between successive partials.}
\]

Our spectral estimator \( U(f) \) relies on maximizing a scalar product between the parametric amplitude spectrum and theoretical harmonic unitary patterns of \( F_0 \) candidates:

\[
U(f) = \sum_{h=1}^{H_f} w_{J,h} S(h_f)
\]

where \( H_f \) is the maximum number of partials possible for fundamental frequency \( f \) and \( \{w_{J,h}, h \in [1, H_f]\} \) is the pattern associated to \( f \). The choice of the pattern is based on an approximative logarithmic spectral decrease of components. The slope \( p \) of a linear

\[
\beta(f_0) = \frac{1 - \beta (f_0)}{1 + \beta (f_0)}
\]
regression between $\log(\sqrt{E_k})$ and $f_k$ is extracted and weights $w_{f,h}$ are then defined as:

$$w_{f,h} = w_0 e^{phf}$$

where $w_0 = \left(\sum_{h=1}^{H_f} e^{phf}\right)^{-\frac{1}{2}}$ is a normalizing term such that $\sum_{h=1}^{H_f} w_{f,h}^2 = 1$.

The spectral estimator is then adapted to piano tones by selecting the values of the spectrum on an inharmonic stretched scale instead of a harmonic scale:

$$U_{inh}(f) = \sum_{h=1}^{H_f} w_{f,h} S\left(hf \sqrt{1 + \beta(f)} (h^2 - 1)\right)$$

Finally, the estimator efficiency can be improved by ignoring all frequencies and weights below a cut-off frequency of 100 Hz since the impedance at the piano bridge [13] causes a significant deviation of low frequencies from the inharmonicity law and the highest weights $w_{f,h}$ of patterns are allocated to those frequencies.

### 2.2.3. Pitch estimator

As mentioned in the introduction, sub-harmonic and harmonic error trends are opposed in temporal and spectral methods. A way to benefit from this phenomenon is described in [8]. It consists in multiplying a temporal and a spectral function on a common $F_0$ scale in order to preserve common peaks from both functions and to remove or attenuate other peaks (see fig. 2). Thus, the pitch is estimated by maximizing the product of the methods $R_{inh}(\frac{1}{f})$ and $U_{inh}(f)$:

$$\hat{f}_0 = \arg\max_f \left(R_{inh}(\frac{1}{f}) U_{inh}(f)\right)$$

Thanks to the analytic expressions (6) and (10), $R_{inh}(\frac{1}{f})$ and $U_{inh}(f)$ can be directly evaluated for any $f$ value. As the $F_0$ distribution of an equal-tempered musical scale is logarithmic, the $F_0$-scale support is set to $N_f$ points logarithmically spaced in the $F_0$-search range. This unconstrained choice is a key advantage of the method since the logarithmic $F_0$ distribution is not offered by many methods (see [4, 8]). Actually, temporal methods have a linearly distributed time scale, which results in a lack of precision in high frequency and too much resolution in low frequency, whereas Fourier-based spectral methods have a linear $F_0$ distribution. In those cases, the estimation function must often be interpolated to achieve the required precision and may still suffer from this.

In a Matlab implementation on a 2.4GHz-CPU, the overall processing of a 60ms frame averages 6.5s. About 1s is necessary for the analysis. About 95% of the remaining time is required by the spectral $F_0$ estimator and may be optimized and written in C for a computationally-efficient implementation.

### 3. EVALUATION

The algorithm has been evaluated on isolated piano tones from various sources: 3168 notes from three pianos of RWC database [14], 270 notes from five pianos of a PROSONUS database and 264 notes from a Yamaha upright piano of a private database. All recordings include several takes of all the 88 notes of piano range (except PROSONUS in which notes are spaced by fourth) with a varying loudness. RWC samples also offer various play modes (normal, staccato, with pedal). The $F_0$ search scale is composed of $N_f = 8192$ values logarithmically distributed between $f_{min} = 26.73$ Hz and $f_{max} = 4310$ Hz. The estimation is performed after the analysis of a single 60 ms or 93 ms frame: 60 ms is quite a challenging frame length since it is below twice the period of lowest notes while 93 ms is a well spread duration for this kind of evaluation. Each estimated $F_0$ is associated to the closest note in the equal tempered scale with A4 tuned to 440 Hz. Errors are then defined as incorrect note estimations. The method is compared to two estimators. The first one is as similar to our estimator as possible, replacing the ESPRIT analysis stage with a classical analysis: the ACF is estimated from the signal by the formula $r(\tau) = \frac{N_s}{N_s-\tau} \\text{DFT}^{-1} \left[|\text{DFT}[s]|^2\right]$, the factor $\frac{N_s}{N_s-\tau}$ being a correction of the bias; the spectral estimator $U_{inh}(f_0)$ is computed by replacing the parametric spectrum with the modulus of the DFT of the signal, using a zero-padding on $8N_f$ points; $r(\tau)$ is mapped to the frequency scale by interpolation as described in [8]; the pitch is finally estimated by maximizing the product between the spectral function and the remapped $r(\tau)$. The second method is the YIN algorithm [4] which is considered as a very efficient monophonic estimator. We used the code available on the authors’ website.

Evaluation results are reported in fig. 3. At the target window length of 60 ms, the global error rate of our estimator is around 4.4%, which is at least twice better than the other estimators. This is due to a low error rate on a large $F_0$ range (1.1% in the $F_0$ range 65 – 2000 Hz) and slowly increasing values at the very bass and treble limits. In comparison, the non-ESPRIT based estimator achieves
a 1.1% error rate in the range 240 – 2000 Hz. Its low efficiency outside this range shows how the \( F_0 \) estimation is improved by both the high resolution analysis and the handling of parametric, analytic formulas. The YIN algorithm is slightly less efficient in the medium range than our estimator and has similar results in the bass range (for the first octave both curves should be at the same level, but our estimator results seem to be worse since they include the lowest four note error rates that cannot be estimated by the YIN algorithm with a 60 ms window length). In the high range, it shows a quite high error rate, which is a typical behavior of temporal methods. Global results are improved with a 93 ms frame length. Nevertheless, the high resolution analysis does not enhance significatively the \( F_0 \) estimation even if its error rate remains the lowest.

Typical errors are now discussed, in the 60 ms analysis case. As expected, usual errors are under-estimations of high \( f_0 \)s and over-estimations of low \( f_0 \)s. Around 18% of errors made by each algorithm are octave and suboctave errors. In the case of our algorithm, the remaining error intervals are of all kinds, with only 5% that are half-tone errors, whereas this rate reaches 10% for the other two algorithms. The YIN algorithm makes a high proportion of sub-harmonic errors (13% are sub-octaves, 8% are sub-nineteenth). Thus, even if our algorithm makes a reduced number of harmonic/subharmonic errors, those errors remain difficult to avoid. Half-tone error rates show the efficiency of our method while the other algorithms suffer from a lack of precision of temporal estimators for high \( F_0 \). Finally, the inharmonicity management contributes to lower the global error rate, from 4.9 to 4.4% in the 60-ms frame case. As expected, the improvement is localized in the lowest \( F_0 \) range: the error rate in the MIDI range [21, 37] decreases from 16.0 to 14.1%.

4. CONCLUSIONS

The \( F_0 \) estimator designed in this paper allows to address typical error trends in a short frame analysis and a wide \( F_0 \)-range context. It is based on a preliminary extraction of the parameters of components and on the design of temporal and spectral parametric function. Satisfying performances have been obtained and a large part was allocated to the discussion on typical errors and the way to avoid them.

5. REFERENCES