A2.11

ESTIMATION OF FUNDAMENTAL FREQUENCY OF MUSICAL SOUND SIGNALS

Boris DOVAL[+] & Xavier RODET[+*]
[+] LAFORIA- Université Paris VI, Paris, France.
Fax: 44 27 62 86. e.mail: doval@laforia.ibp.fr
[*]IRCAM, Paris, France.
Fax: 42 77 29 47. e.mail: rodet@ircam.ircam.fr

ABSTRACT

In order to realize an estimation of the fundamental frequency (f0) of pseudo-periodical sounds with a wide band of possible f0, we have set up a theoretical model based on a maximum likelihood for f0. Then we have simplified our model so as to make it fast enough for extensive tests. We have tested the resulting algorithm on musical and speech sounds and presented the results. As a musical application, we have implemented an instrument follower based on our algorithm and which operates in real time.

INTRODUCTION

We present a new method for estimation of fundamental frequency (f0) of musical sound signals. It operates on a large interval of f0 values (typically from 50 to 4000 Hz) with a short delay (typically less than 20 ms) and as small an error rate as possible.

Knowledge of f0, sometimes misnamed pitch, is useful in many cases: in acoustics, to synchronize signal analysis algorithms, for analysis-synthesis applications, for real time instrument following, etc.

Generally existing methods are not very satisfactory: some methods have been developed for speech (cepstrum, autocorrelation,...) and are inadequate to musical signals, particularly because of the large range of f0 values or the variety of spectra encountered. Other methods have been developed especially for music, but commercially available devices usually have too high error rates and too long response delays after note onset.

We think that much better results can be obtained if an elaborate enough algorithm is designed which can be applied to music and speech sounds without restriction.

A signal may exhibit several periodicities none of which is outstanding; therefore our algorithm should propose several f0 values with corresponding weights (the estimation of several notes in a chord is also a possibility). Moreover, estimation of f0 cannot be done independently of some specific context (for instance, within a given interval) or of a given application; consequently it is important that the method be flexible enough to take into account a priori knowledge other than the acoustic signal alone.

PROBLEM DESCRIPTION

One of the motivations of our research is that in many cases where usual algorithms are erroneous the *correct* value of f0 is *obvious* to the user looking at the signal window or the Short Term Fourier Transform (STFT). Our goal is somehow to design an algorithm that would give a correct value each time a human observer is not confused. Otherwise the notion of periodicity is probably meaningless for the analyzed window.

This paper deals with the estimation of f0 taken in the sense of the signal periodicity in a short window (for instance less than 40 ms). We are not looking for *perceived pitch* but for the optimal period duration(s) in the window according to a criterion to be defined. This also means that we are postponing possible error corrections as obtained by considering several successive windows. It seems necessary to first optimize the results on a single window before turning to higher levels of processing. Finally, for the low level stage described here we propose a general purpose algorithm excluding any learning process performed beforehand on the class of the analyzed signal (an algorithm using a learning stage is described in [1]).

THEORY

Supposing that the harmonic partials in the signal are known, our hypothesis is that the information about f0 is to be found in this set of partials. So we are looking for the f0(s) whose harmonics best *'explain'* the signal's partials. An approximation of this set of partials can be obtained from the STFT, for instance by extracting the maxima m_i of the modulus of the STFT or by locating the plateaus of the instantaneous frequency spectrum.

Actually the set $M=\{m_i\}$ of so-called *partials* found in the STFT not only contains harmonic partials of the f0 being sought but also supplementary components. These supplementary components either correspond to non-harmonic partials appearing in the signal, or to non-sinusoidal components that we call *noise* . Similarly some harmonics of the target f0 may be absent and thus not represented in M.

Let us call φ_i the frequency of m_i, a_i its amplitude and f0 an estimation of the fundamental frequency. Therefore in following developments the probabilities are conditional to the value f0. Only the m_i and the harmonic partials of f0 whose frequencies are less than Fmax will be considered, where Fmax is a frequency limit depending on the context and inferior to half the sampling rate. We have to decide which m_i represent harmonic partials and which do not. This procedure could be called matching the observed maxima $\{m_i\}$ with the theoretic series of f0's harmonic partials. Let I_k be the interval $[(k-0.5).f0,\ (k+0.5).f0]$. The k^{th} harmonic of f0 can be considered to be taking its value in I_k. Therefore only the φ_i belonging to I_k can represent the k^{th} harmonic. Let M_k be the set of partials m_i whose frequency φ_i belongs to I_k. The set of the M_k is a partition of the set M of partials. Let E_k be the event that the k^{th} harmonic is present and $\underline{E}_k$ be the opposite of E_k. Then

$$P(M_k) = P(M_k / E_k) \,.\, P(E_k) + P(M_k / \underline{E}_k) \,.\, P(\underline{E}_k) \quad \mathbf{(1)}$$

Let us call K the set of k for which M_k is not empty and $\underline{K}$ the set of k for which M_k is empty. If k belongs to $\underline{K}$ then $P(M_k)$

CH2977-7/91/0000-3657 $1.00 © 1991 IEEE

is equal to $P(M_k / \underline{E}_k).P(\underline{E}_k)$ since the probability that the k^{th} harmonic is present while there is no partial is equal to zero. If k belongs to K, in the first term of **(1)**, $P(M_k / E_k)$ can be broken down as follows:

$$P(M_k / E_k) = \sum_{m_j \in M_k} g(\varphi_j/f0 - k)\ h(a_j)\ Ps(M_k - \{m_j\})$$

where g implements the distribution of the k^{th} harmonic (for example a Gaussian [2]) around the frequency k*f0, where h is the probability that the amplitude of the k^{th} harmonic be a_j, where $M_k - \{m_j\}$ is the set M_k without the partial m_j, and where $Ps(M_k - \{m_j\})$ is the probability to observe the $M_k - \{m_j\}$ distribution of supplementary components; this latter probability can be estimated in terms of the total number of supplementary partials observed, that is $Card(M) - | Fmax/f0 |$. In the second term of **(1)**, still in the case where k belongs to K, we can use the equality:

$$P(M_k/\underline{E}_k) = Ps(M_k)$$

Let us recall that we want to calculate the likelihood of f0, L(f0), that is the probability of the observation M conditional to f0. Therefore supposing that the M_k distributions are independent of each other, we can write

$$L(f0) = \prod_{k} P(M_k)$$

$$= \prod_{k \in K} \Big(P(E_k) \sum_{m_j \in M_k} g(\varphi_j/f0 - k)\ h(a_j)\ Ps(M_k - \{m_j\}) + Ps(M_k)\ P(\underline{E}_k) \Big) . \prod_{k \in \underline{K}} Ps(M_k)\ P(\underline{E}_k) \quad \textbf{(2)}$$

We then would have to choose the f0 value which best represents the observation according to the criterion of the maximum likelihood. Since the mathematical expression for L is rather complex, in order to simplify it, the few greatest terms only can be substituted for the sum on $m_j \in M_k$. They are obtained from the m_j partials with a frequency close to k*f0 and an amplitude which fits the spectrum envelope. This approximation is validated by the fact that we then only neglect harmonic partials with a weak amplitude or a very inharmonic frequency. Notice that taking only the greatest term in the sum on $m_j \in M_k$, obtained for $j=j_k$, the f0 which maximizes L maximizes also:

$$\log(L(f0)) = \sum_{k \in K} \log(P(E_k)) + \sum_{k \in \underline{K}} \log(P(\underline{E}_k))$$
$$+ \sum_{k \in K} \log(Ps(M_k - \{m_{j_k}\})) + \sum_{k \in \underline{K}} \log(Ps(M_k))$$
$$+ \sum_{k \in K} \log(g(\varphi_{j_k}/f0 - k)) + \sum_{k \in K} \log(h(a_{j_k}))$$

Notice also that if we choose the g distribution to be Gaussian, then we can write the corresponding term as follows:

$$- \sum_{k \in K} \log(\sigma \sqrt{2 \pi}) - (1/(2.\sigma^2.f0^2)) \sum_{k \in K} (\varphi_{j_k} - k*f0)^2$$

We then observe that $\Sigma (\varphi_{j_k} - k*f0)^2$ corresponds to the mean square distance between the vector of the φ_{j_k} and the vector of the k*f0.

However, without an analytical solution of **(2)**, we have to compute and compare L(f0) for all possible f0 values.

IMPLEMENTATION

Preprocessing of the current window consists in extracting maxima of the magnitude STFT. It is also possible to consider the spectrum values around a maximum in order to compute a probability that the maximum corresponds to a partial rather than to noise [3].

a. Frequency axis partition

According to the theory one would have to examine all possible f0 values and compare their likelihoods. However, the optimal solution can be found by examining only a limited number of intervals properly selected on the frequency axis.

All f0s which have an equal number of harmonic partials inside the frequency band [0, Fmax] can fall in the same interval. An equivalent condition is that a given partial of frequency φ_i should not correspond at the same time to different harmonic numbers for different f0 values in the same interval, which can be written: for any φ_i and for any k, φ_i/k and $\varphi_i/(k+1)$ should not belong to the same interval. To fulfill this requirement one can use a hyperbolic scale with equation n=Fmax/f(n) where n is the interval number and f(n) its central frequency.

Thus, to find the optimal solution we proceed in two steps: first we determine the interval which contains the optimal solution, then we compute the precise optimal value for f0 within this interval.

b. Computation of the histogram

To determine the interval which contains the optimal solution, we construct a histogram on the selected intervals of the frequency axis by computing the value of the likelihood L(f0) for each interval n. The computation of this histogram places our algorithm in the class of *harmonic matching frequency domain Pitch Detector Algorithms* [4].

c. Heuristics

The maximum values in the histogram allow for the selection of the most probable intervals. Then more precise optimal f0 values are computed by using a regression on the frequencies of the signal partials which are matched with f0 harmonic partials.

A certain number of heuristics are also applied, particularly to avoid octave errors. They use criteria based on the amplitudes of partials weighted by their harmonicity and on the regularity of the amplitude envelope.

d. Real time imp!ementation

The algorithm has also been implemented on the *musical workstation* at IRCAM [5] in order to follow the pitch of notes played by an instrument. With the parameter values given below the program operates in *real time*, the computing time for a window being less than the time between two successive windows.

ALGORITHM VALIDATION AND EVALUATION

Our algorithm has been tested on several musical and speech sounds, and it has been found to give correct values, with errors only limited to the unavoidable ones. For instance, it is clear that such an algorithm using a unique signal window may be undetermined on the frontier between two notes.

a. Test on musical sounds

John Kitamura [6] has associated our algorithm with an algorithm quantifying frequencies into notes of the scale and detecting beginning and ending of notes (note-ons and note-offs). The output of these two algorithms in series is a musical event (MIDI event) sequence which is easily comparable to the score played by the instrumentist (Figure 1).

An example is given in Figure 2. The first line is an extract of the score [10] which was played by the clarinettist. The second line is the raw amplitude output superimposed with the note-on and note-off information. The third line is the continuous f0 output superimposed with stabilized MIDI events. The final line shows the MIDI note-on events, which should correspond to the original score. The stems were added manually in order to be able to make a clear comparison with the original score.

The application operates in real time on an Intel i860. An important feature is the response delay. It is the sum of three terms:

- half the analysis window length, assuming that analysis time is at the midpoint of the window. This term is inherent to the f0 algorithm.
- the duration necessary to warrant the stability of the estimated f0 frequency, in terms of the number of successive windows for which f0 is stable. This parameter of the quantifying algorithm is adjustable, and is set to 2.
- the calculation time.

With the test parameter values, these three terms are respectively equal to 11.6, 10 and 5 ms. An extremely difficult passage could result in about 5% note errors, usually occurring at a noisy attack and lasting for only 20 or 30 ms. Several improvements are under study to decrease this error rate. Increasing the delay by 20 ms (by setting the stability parameter to 4) gives a completely error-free result.

b. Test on continuous speech

In the test on continuous speech at the opposite of the previous test, our algorithm was **not** followed by any postprocessing using several successive frames. The use of such a postprocessing is known to improve the results. But we first want to estimate the result of each frame. We have at our disposal a continuous speech database (40 minutes) the f0 of which has been computed and manually verified. It is an extract of the ATR (Advanced Telecommunication Research Institute) Japanese speech database [7,8,9]. The sampling rate is 12 kHz and the frame rate is 400 Hz. The f0 values given by our algorithm have been compared to the 'correct' database values for 20 sentences. The test parameter values were:

- window length: 40 ms
- lower and upper f0 search limits: 40 - 400 Hz
- Fmax: 1200 Hz

At this setting the gross error rate is about 6%, most of errors occuring at end of sentences where the frequency is low (f0 < 80 Hz), which can be easily understood since the analysis window size should be at least 3 periods.

FUTURE RESEARCH

The next step in our research is to determine the different fundamental frequencies for a multiphonic sound. This of course requires a more complex treatment using, among others, multiresolution analysis. Then, apparition and disparition of partials can be detected. Partials can be grouped according to their harmonicity and time behavior correlation in order to find the set of optimal f0 values.

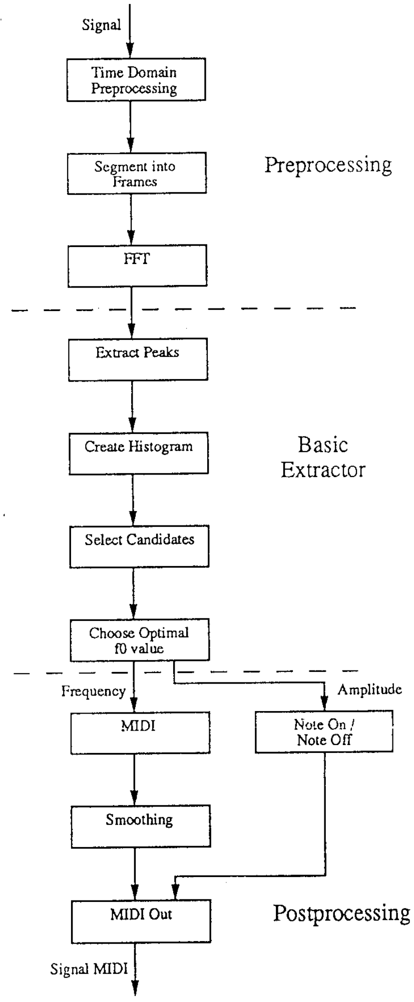

Figure 1 - Block Diagram of the Algorithm Structure

CONCLUSION

We have shown that it is possible to achieve an estimation of the fundamental frequency (f0) of pseudo-periodical sounds with a wide frequency band. The algorithm based on a maximum likelihood for f0 has been successfully tested - especially for music sounds - and was proved to operate in real time with a small error rate and a short delay. It is being used at IRCAM to follow extremely complex and rapid monophonic melodies.

AKNOWLEDGEMENTS

We want to aknowledge A. De Cheveigné and ATR Interpreting Telephony Research Laboratories who let us use the ATR Japanese speech database, and J. Kitamura and M. Puckette for their collaboration to the test on musical sounds.

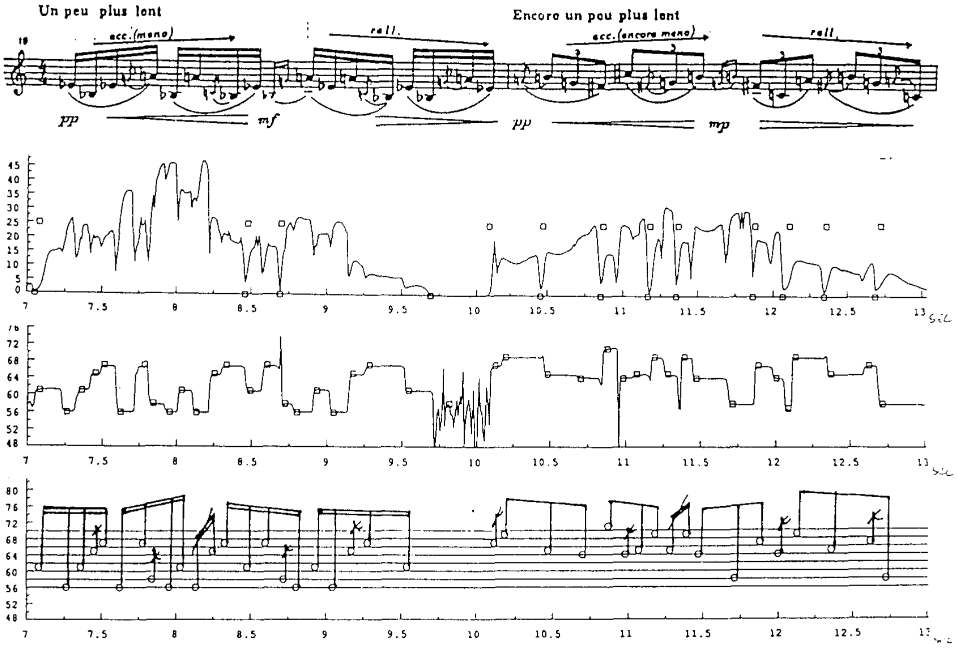

Figure 2 - Example of MIDI output
Original Score, Raw Amplitude, Raw Frequency and MIDI output

REFERENCES

[1] Thouard J.P., Depalle P., Rodet X. (1990)
Detection of pitch of musical sounds with a neural network, INNC, Paris.

[2] Goldstein J. (1973)
An optimum processor theory for the central formation of the pitch of complex tones, J. Acoust. Soc. Amer. 54, 1496-1516.

[3] Rodet X., Depalle P., Poirot G., (1987)
Speech Analysis and Synthesis, European Conf. on Speech Technology, Edinburgh, U.K., Sept 87.

[4] Hess W. (1983),
Pitch determination of speech signals/ algorithms and devices, Springer Verlag, Berlin.

[5]Lindeman E. (1990),
The IRCAM Musical Workstation: Hardware Overview and Signal Processing Features, ICMC 1990 Proceedings, Glasgow.

[6] Kitamura J., Doval B., Rodet X. (1990),
A robust pitch-tracker, to be published in Computer Music Journal.

[7] Kuwabara H., Sagisaka Y., Takeda K., and Abe M. (1989),
Construction of ATR Japanese speech database as a research tool, ATR technical report, TR-I-0086.

[8] Abe M., and Kuwabara H. (1989),
Pitch frequency database on continuous speech, ATR technical report TR-I-0078.

[9] Cheveigné A. (1990),
Experiments in pitch extraction, ATR technical report TR-I-0138.

[10] Boulez P. (1989),
Dialogue de l'ombre double, Universal Edition A. G. Wein.