Multipitch detection

Gaël Richard, Roland Badeau

The aim of this practical work (TP) is to program a multipitch detection algorithm of a musical signal (or equivalently to determine the different fundamental frequencies - or musical notes - of a sound excerpt). Several methods exist, and we propose here to study a simple algorithm inspired by the work of A. Klapuri [1, 2].

This method includes the following steps :

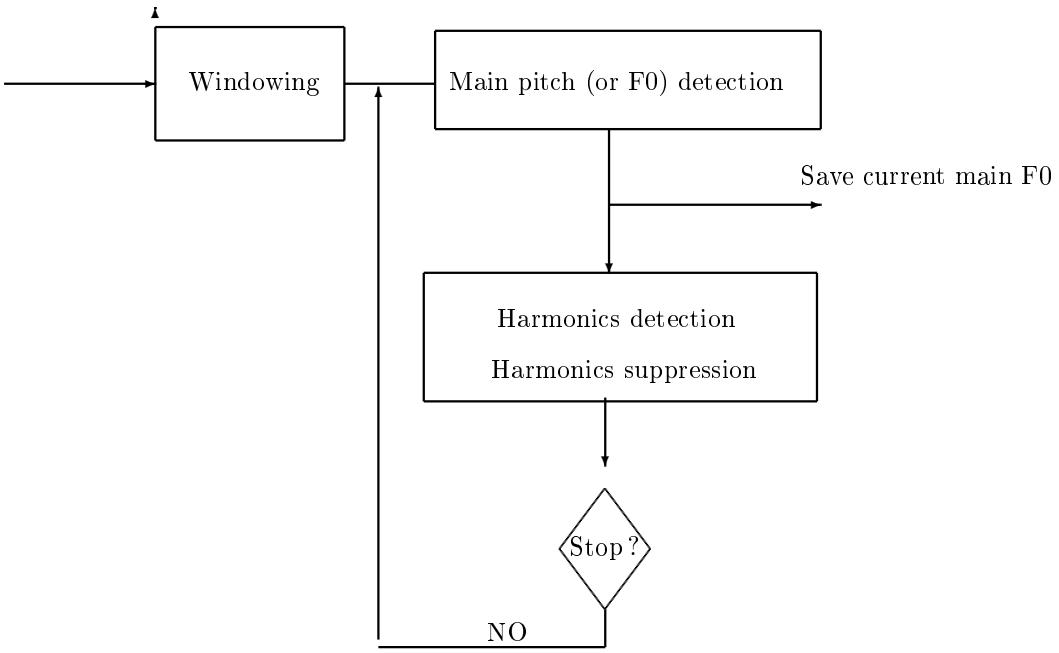
- Estimate the dominant fundamental frequency
- Deduce the location and amplitude of all its harmonics on the signal spectrum
- Subtract the contribution of the corresponding sound (e.g. the sound made of the detected fundamental frequency and its harmonics)

The process is then iterated until all notes have been detected.

In a second step, some refinements or options are proposed. Optionally, it is also suggested to evaluate the spectral smoothness principle which will be applied prior to the spectral subtraction. The performances of these approaches will be assessed on different chords composed of notes played by different instruments (piano, flute and oboe).

This TP will be realized in `MATLAB` or Python. A template named `TPmultipitch4_template.m` for the program can be downloaded on the site mentioned above (for Matlab only)

The figure below summarizes the different steps of the algorithm :



1 Windowing and Fourier transform

Study the first part of the template (`TPmultipitch4_template.m`).

1. What is the window size used ? Is it an appropriate size ? justify your answer
2. What is the purpose of the offset parameter ?

3. What is the precision in Hertz of the spectral representation obtained after Fast Fourier Transform (FFT) ?

2 Fundamental frequency estimation by the spectral sum method

We will estimate the fundamental frequency F_0 of a signal x of length N, sampled at frequency F_s. The value of F_0 will be estimated in the range $[F_{\min}, F_{\max}]$, with a precision of at least dF. We will first use the spectral sum method which consists in adding H compressed versions of the spectrum. The spectral sum is then given by :

$$S(e^{2j\pi f}) = \sum_{n=1}^{n=H} |X(e^{2j\pi nf})| \tag{1}$$

The parameters of our estimator will then be :
- dF_{min}, the minimal frequency resolution (by default $dF_{min} = Fs/N$ where N is the window size) ;
- F_{min} and F_{max} provide the search interval for the different fundamental frequencies (typical values could be $F_{\min} = 50$ Hz, $F_{\max} = 900$ Hz)
- `H`, the number of compressed versions of the spectrum (Typical value : $H = 4$)

2.1 Understanding the computation of the signal spectrum *(see template)*

The Discrete Fourier Transform of the signal x is computed on N_{fft} points. This is done by first multiplying x by a hamming window of size N to decrease the height of the side lobes. What is the minimal value for N_{fft} to obtain a precision of at least dF_{min} (a power of 2 will be chosen in order to use the fast FFT algorihm). For this the function `nextpow2` can be used.

2.2 Computation of the spectral sum

The spectral sum will be saved in a vector S of dimension R.
- What is the maximum frequency that can be used for the computation of S (as a function of H, R and Fs) ?
- What is then the maximum value of R that can be computed (while being below the Nyquist frequency) ?
- Compute the spectral Sum S.

2.3 Finding the spectral sum maximum

Find the integer value $N_{\min}$ and $N_{\max}$ which correspond to the range $[F_{\min}, F_{\max}]$ (note that $N_{\max}$ should be less than R). Localize the maximum of S on the interval $[N_{\min}, N_{\max}]$, and deduce the value of the fundamental frequency. The method will be evaluated on monophonic signals such as `A4_piano.wav ou E4_oboe.wav.`

3 Subtraction of the sound corresponding to the detected fundamental frequency

3.1 Harmonics detection

It is first necessary to detect the different harmonics of the fundamental frequency F_0. To that aim, it is proposed to search for the maximum around each theoretical harmonic of frequency $f_k = k.F_0$. One may choose for example $f_{kmin} = (1 - \alpha) * f_k$ and $f_{kmax} = (1 + \alpha) * f_k$. (Note that this search can also be applied to F_0 to obtain a more accurate estimation).
- What is an appropriate choice for α ?

– Is this process of harmonics search appropriate ? for all types of music instruments ?

3.2 Harmonics suppression

Once detected, the harmonics of the detected harmonic sound (or note) must be suppressed from the recording. It is first necessary to estimate the theoretical width of a spectral peak which is dependent of the analysis window size. Then, all frequency bins of each harmonic are forced to zero (e.g. all frequency bins in the range $[k1 : k2]$ which define the spectral width of a spectral peak). Note that, it may be preferable to force the value of the frequency bins of an harmonic to the minimum value of the spectrum on this interval such that $|X_k(k1 : k2)| = \min(|X_k(k1 : k2))|$; (this corresponds to force the amplitude of the harmonic to the level of the background noise).

3.3 Stopping criterion

All steps described above are then iterated until a predefined stopping criterion is reached.
– Find an appropriate stopping criterion which allows to automatically detect when all notes were found.
The algorithm will be tested on the different chords provided.

4 Some refinements or options

4.1 A different F_0 detector

As an alternative, the spectral product may be used instead of spectral sum given in equation 1. With this new estimator it is mandatory to force the value of a supressed harmonic to the level of the noise background (see section 3.2).
– Compare the performances of the two estimators on monophonic signals and chords.

4.2 Taking account of inharmonicity

Some instruments, such as piano for example are not perfectly harmonic. For such instruments, it may be necessary to take this into account for localizing and then suppressing the harmonics. This may be done by changing the search interval $[f_{kmin} : f_{kmax}]$ for each harmonic by defining $f_{kmin} = (1 - \alpha) * f_{inharmo}$ and $f_{kmax} = (1 + \alpha) * f_{inharmo}$ with $f_{inharmo} = k * F_0 * \sqrt{(1 + k^2 * \beta)}$ where β is the inharmonicity coefficient.
– What is an appropriate value of β for a piano sound ? for an oboe sound ?
– Can this coefficient be automatically estimated ?

4.3 Harmonic suppression with spectral smoothness principle (Optionnal)

To better separate musical notes which are in harmonic relation it is preferable to slightly change the suppression step. The spectral smoothness principle consists in estimating to which extent an harmonic should be suppressed. Basically, the aim is to only suppressed the energy of the harmonic of the detected note without affecting the harmonics of other sounds even if they are in the same region. The spectral smoothness principle is to compute a smooth spectrum where the amplitude of each harmonic f_k is replaced by the average amplitude of neighbouring harmonics f_{k-1}, f_k and f_{k+1}. Then, spectral subtraction can be summarized as :
– forcing the spectral value around each harmonic to the minimum in the range $[k1 : k2]$ to $(|X_k(k1 : k2)| = \min(|X_k(k1 : k2))|)$ if the spectrum value is below the smooth spectrum.
– forcing the spectral value around each harmonic to the difference between the spectrum and smooth spectrum in the range $[k1 : k2]$ if the spectrum value is above the smooth spectrum.

Références

[1] A.P. Klapuri. Multipitch estimation and sound separation by the spectral smoothness principle. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, May 2001.

[2] A.P. Klapuri. Multiple fundamental frequency estimation by harmonicity and spectral smoothness. *IEEE Trans. Speech and Audio Processing*, 11(6) :804–816, 2003.