MULTIPLE FUNDAMENTAL FREQUENCY ESTIMATION BASED ON SPECTRAL PATTERN LOUDNESS AND SMOOTHNESS

Antonio Pertusa, José M. Iñesta
University of Alicante, Spain
Departamento de Lenguajes y
Sistemas Informáticos

ABSTRACT

Two multiple fundamental frequency estimation systems are presented in this work. In the first one (PI1, PI2), the best fundamental frequency candidates combination is found in a frame-by-frame analysis by applying a set of rules, taking into account the spectral smoothness measure described in this work. The second system (PI3) was used to extract symbolic features for audio genre classification in a fast way, so the evaluation of this system can reveal the potential of another similar approaches to support these kind of tasks.

1 INTRODUCTION

The goal of a music transcription system is to extract a score from an audio piece. A multiple fundamental frequency estimator is the main piece of a polyphonic transcription system, whereas tempo detection and key estimation complement it to correctly extract the score. Two multi-f_0 estimators are presented in this work.

The first system takes into account the smoothness and the amplitudes of the harmonics of each f_0 candidate, performing a frame by frame analysis.

The second one has been introduced in [2], where the goal was to improve previous music genre classification results by extension of the feature space including features extracted from symbolic data. This system was intended to be a fast prototype to extract the notes from an audio file. In general, multi-f_0 estimation is a very difficult task, but in spite of the amount of false positive and negative notes that these kind of systems produce, the results for genre recognition increased when adding symbolic features extracted using this multi-f_0 estimator.

2 SYSTEMS DESCRIPTION

2.1 Multi-f_0 estimation using Gaussian smoothness

This system uses Gaussian spectral smoothness as salience measure to select f_0 candidates. This estimator converts a mono audio file sampled at 44 kHz into a sequence of notes. First, it analyzes the target song performing a Short Time Fourier Transform (STFT). To compute it, a Hanning window with 4096 samples and 50% overlap has been used. With these parameters, the temporal resolution is 46 ms. Zero padding has been used, multiplying the original size of the window by 4 and adding zeroes to complete it before the STFT is computed. This technique does not increase resolution, but the estimated amplitudes and frequencies of the new spectral bins are usually more accurate than applying interpolation.

In order to adapt the system to the MIREX frame-by-frame evaluation requirements, the overlapping percentage was changed to be 90%, getting a temporal resolution of 9.28 ms. As the window size is large, this is a drawback, because temporal precision don't increment much the detection results and it makes the system slower.

To detect the fundamental frequencies in the target frame, a set of f_0 candidates is selected first. A spectral peak is a candidate if it's within the range [38 Hz , 2100 Hz], and at least two of its harmonics are found. To search for harmonics, a fixed range [-10 Hz , 10 Hz] around each harmonic frequency hf_0 for $h = 2, 3, ...$ is considered. The peak which is closest to this frequency within this range is set as a harmonic partial, and if there are no peaks in this range then that harmonic amplitude is set to 0.

Candidates are ordered by the sum of their harmonic amplitudes and, as maximum, only the first C candidates of this list ($C = 10$) are considered. Then, all the possible candidate combinations (chords) are calculated, and the chord with best salience will be chosen in the target frame.

A candidate salience is computed by taking into account the loudness and smoothness of its harmonic amplitudes. To get these values, an iterative algorithm is applied. First, for each candidate, harmonics are searched and their amplitudes are stored in a vector. Then, each harmonic is marked with a label containing all the candidates that the harmonic belongs to. From the lowest to the highest frequency candidate, the harmonic amplitudes stored in the amplitude vectors are updated; for each candidate, the non-shared harmonic amplitudes stay the same, but the shared harmonic amplitudes are linearly interpolated using the non-shared amplitudes in the same candidate vector. If an interpolated value is greater than the obtained harmonic amplitude, then the candidate's harmonic value in the vector will remain the same and the spectral peak will be removed for other candidates. If the inter-

polated value is smaller, this value will be assigned to the candidate harmonic vector and will be subtracted from the corresponding spectral peak.

When this process is done for all the candidates in a combination, each candidate loudness l is computed by summing all the values of its amplitude vector. Smoothness is also computed for each vector, by using a Gaussian filtering; the idea is that a smooth spectral pattern should be more probable than a sharper one. To compute the smoothness of a harmonic amplitudes vector, **h**, it is low-pass filtered using a truncated Gaussian window with three components $G_{\sigma=0.5} = \{0.2, 0.6, 0.2\}$, that is convolved with **h** obtaining the smooth version, $\tilde{\mathbf{h}} = G_{0.5} * \mathbf{h}$. Then a *sharpness* measure is computed as $S = \tilde{\mathbf{h}} - \mathbf{h}$. This value is normalized, $\bar{S}$, and the smoothness is set as $s = 1 - \bar{S}$.

Once the smoothness and the loudness of each candidate have been calculated, the salience of a note is computed as $l \cdot s^2$, and the salience of a combination of notes is the sum of all its note saliences. The combination with best salience is the winner chord in this frame. Combinations that have at least one candidate with $l < 0.1L$ are discarded, being $L = \max\{l\}$ the loudest candidate.

After selecting the f_0 candidates in all the frames, a last stage is applied to avoid local errors. If a given frequency was not detected in a target frame but it was found in the previous and next frames, it is considered to be detected in the current frame too, avoiding discontinuities in the detection. Finally, very short notes (less than 6 frames, i.e. 55.68 ms) are removed, and the sequences of consecutive detected fundamental frequencies are converted to MIDI notes.

2.2 Multi-f_0 estimation using constant spectral pattern matching

This system was integrated within the audio genre classifier proposed in [2]. As part of a more complex system, it is very important for this multi-f_0 estimator to be fast. To achieve it, firstly the STFT is computed using the same parameters than in the previous implementation, but only those frames after onsets are computed to detect the pitches.

After the STFT, the onset detection stage described in [3] is performed, classifying each time frame t_i as onset or not-onset. The system searches for notes between two consecutive onsets, analyzing only one frame between two onsets to detect each chord. To minimize the note attack problems in f_0 estimation, the frame chosen to detect the active notes was $t_o + 1$, being t_o the frame where an onset was detected. Therefore, the spectral peak amplitudes computed 46 ms after an onset provide the information to detect the chord.

For each frame, we use a peak detection and estimation technique proposed by X. Rodet in [5] called Sinusoidal Likeness Measure (SLM). This technique can be used to extract spectral peaks corresponding to sinusoidal partials, and this way residual components can be removed; this makes sense for songs with percussive instruments (like most popular music). SLM needs two parameters; the bandwith W, that has been set as $W = 50$ Hz and a threshold $\mu = 0.1$. If the SLM value for a peak $v_\Omega < \mu$, it will be removed. After this process, an array of sinusoidal peaks for each chord is obtained.

Given an array of spectral peaks, we have to estimate the pitches of the notes. First, the f_0 candidates are chosen. This selection depends on their amplitudes and their frequencies. If a spectral peak amplitude is lower than a given threshold (experimentally, 0.05 reported good results), the peak is discarded as f_0 candidate, because in most instruments usually the first harmonic f_0 has a high amplitude. There are two more restrictions for a peak to be a f_0 candidate; only f_0 candidates within the range [50 Hz , 1200 Hz] are considered, and the absolute difference in Hertz between the candidate and the pitch of its closest note in the well-tempered scale must be less than f_d Hz. Experimentally, setting this value to $f_d = 3$ Hz yielded good results. This is a constant value independent of f_0 because this way many high frequency peaks that generate false positives are removed.

Once a subset of f_0 candidates is obtained, a constant spectral pattern is applied to determine whether the candidate is a note or not. The spectral pattern used in this work is a vector in which each position represents a harmonic value relative to the f_0 value. Therefore, the first position of the vector represents the f_0 amplitude and it will always be 1, the second position contains the relative amplitude of the second partial respect to the first one and so on. The spectral pattern sp used in this work contains the amplitude values of the first 8 harmonics, and it has been set as $sp = \{1, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.01\}$, which is similar to that proposed by A. Klapuri in [1]. As different instruments have different spectra, this general pattern is more adequate for some instruments, as a piano, and less realistic for others, like a violin. This pattern was selected from several combinations tested.

An algorithm is performed over all the f_0 candidates to determine whether a candidate is a note or not. First, the partials that are whole multiples of each f_0 candidate are found. A harmonic h of f_0 is found when the closest spectral peak to hf_0 is within the range $[2hf_0 - f_h, f_h]$ Hz, being f_h:

$$f_h = hf_0\sqrt{1 + \beta(h^2 - 1)} \quad (1)$$

with $\beta = 0.0004$. There is a restriction for a candidate to be a note; a minimum number of its harmonics must be found. This number was empirically set as half the number of harmonics in the spectral pattern. If a candidate is considered as a note, then the values of the harmonic amplitudes in the spectral pattern (relative to the f_0 amplitude) are subtracted from the corresponding spectral peak amplitudes. If the result of a peak subtraction is lower than zero, then the peak is removed completely from the spectral peaks. The loudness l_n of a note is the sum of its expected harmonic amplitudes.

After this stage, a vector of note candidates is obtained at each time frame. Notes with a low absolute or relative loudness are removed; firstly, the notes with a loudness

$l_n < \gamma$ are eliminated. Experimentally, a value $\gamma = 5$ reported good results. Secondly, the maximum note loudness $L = \max_n\{l_n\}$ at the target frame is computed, and the notes with $l_n < \eta L$ are also discarded. After some experiments, $\eta = 0.1$ was eventually chosen. Like in the previous system, the sequences of consecutive detected fundamental frequencies are converted to MIDI notes.

3 EVALUATION

The evaluation was done at two different levels; frame by frame pitch estimation and note tracking. As the constant spectral pattern (PI3) system did not perform a frame by frame analysis (an onset detection stage was used and only the frames after each onset were taken into account), it was only evaluated in the note tracking contest.

The multi-f_0 estimator that uses Gaussian smoothness (PI1) was evaluated in the MIREX frame by frame multi-f_0 estimation contest, whose results are shown in table 1, and the corresponding runtimes are in table 2. It can be seen that the accuracy of the system is close to the highest accuracy of the analyzed systems, being the one with best precision and with lowest E_{tot} error [4]. The difference between precision and recall suggests that maybe too many notes were filtered out, so probably changing the note removal thresholds could yield a higher accuracy. As can be seen in table 2, the performance of the system is very good compared to the other systems analyzed.

id	Acc.	Pr	Re	E_{tot}	E_{subs}	E_{miss}	E_{fa}
RK	0.605	0.690	0.709	0.474	0.158	0.133	0.183
CY	0.589	0.765	0.655	0.460	0.108	0.238	0.115
ZR	0.582	0.710	0.661	0.498	0.141	0.197	0.160
PI1	**0.580**	**0.827**	**0.608**	**0.445**	**0.094**	**0.298**	**0.053**
EV2	0.543	0.687	0.625	0.538	0.135	0.240	0.163
CC1	0.510	0.567	0.671	0.685	0.200	0.128	0.356
SR	0.484	0.614	0.595	0.670	0.185	0.219	0.265
EV1	0.466	0.659	0.513	0.594	0.171	0.371	0.107
PE1	0.444	0.734	0.505	0.639	0.120	0.375	0.144
PL	0.394	0.689	0.417	0.639	0.151	0.432	0.055
CC2	0.359	0.359	0.767	1.678	0.232	0.001	1.445
KE2	0.336	0.348	0.546	1.188	0.401	0.052	0.734
KE1	0.327	0.335	0.618	1.427	0.339	0.046	1.042
AC2	0.311	0.373	0.431	0.990	0.348	0.221	0.421
AC1	0.277	0.298	0.530	1.444	0.332	0.138	0.974
VE	0.145	0.530	0.157	0.957	0.070	0.767	0.120

Table 1. Frame by frame evaluation results.

Both systems were evaluated for the note tracking contest. Despite they were not designed for this task (the analysis is performed using individual frames), their results were satisfactory. As shown in the table 3, the gaussian approach (PI2) doubles the accuracy of the constant spectral pattern system (PI3). These results suggest that replacing the PI3 system by PI2 into the genre classification itinerary proposed in [2] could increase the results of this music classifier. As can be seen in table 4, both systems are the ones with fastest processing times.

id	Runtime	Machine
CC1	2513	ALE Nodes
CC2	2520	ALE Nodes
KE1	38640	ALE Nodes
KE2	19320	ALE Nodes
VE	364560	ALE Nodes
RK	3540	SANDBOX
CY	132300	ALE Nodes
PL	14700	ALE Nodes
ZR	271	BLACK
SR	41160	ALE Nodes
PI1	**364**	**ALE Nodes**
EV1	2366	ALE Nodes
EV2	2233	ALE Nodes
PE1	4564	ALE Nodes
AC1	840	MAC
AC2	840	MAC

Table 2. Frame by frame run times. The first column shows the participant, the second are the running times and the third column is the machine where the evaluation was performed.

	Precision	Recall	Ave. F-measure	Ave. Overlap
RK	0.578	0.678	0.614	0.699
EV4	0.447	0.692	0.527	0.636
PE2	0.533	0.485	0.485	0.740
EV3	0.412	0.554	0.453	0.622
PI2	**0.371**	**0.474**	**0.408**	**0.665**
KE4	0.263	0.301	0.268	0.557
KE3	0.216	0.323	0.246	0.610
PI3	**0.203**	**0.296**	**0.219**	**0.628**
VE2	0.338	0.171	0.202	0.486
AC4	0.070	0.172	0.093	0.536
AC3	0.067	0.137	0.087	0.523

Table 3. Note tracking results based on onset and pitch. Precision, recall, average f-measure and average overlap are shown.

Participant	Runtime (sec)	Machine
AC3	900	MAC
AC4	900	MAC
RK	3285	SANDBOX
EV3	2535	ALE NODES
EV4	2475	ALE NODES
KE3	4140	ALE NODES
KE4	20700	ALE NODES
PE2	4890	ALE NODES
PI2	**165**	**ALE NODES**
PI3	**165**	**ALE NODES**
VE	390600	ALE NODES

Table 4. Note tracking running times. First column is the participant, the second is the running time (in seconds) and the third is the machine where the evaluation was performed.

4 ACKNOWLEDGMENTS

Thanks to Anssi Klapuri for his help and advice. This work is supported by the Spanish PROSEMUS project with code TIN2006-14932-C02.

5 REFERENCES

[1] A. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proc. ISMIR*, pages 216–221, Victoria, Canada, 2006.

[2] T. Lidy, A. Rauber, A. Pertusa, and J.M. Iñesta. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proc. ISMIR*, Vienna, Austria, 2007.

[3] A. Pertusa, A. Klapuri, and J.M. Iñesta. Recognition of note onsets in digital music using semitone bands. In *Proc. 10th Iberoamerican Congress on Pattern Recognition (CIARP)*, LNCS, pages 869–879, 2005.

[4] Graham E. Poliner and Daniel P. W. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 48317, 9 pages, 2007. doi:10.1155/2007/48317.

[5] X. Rodet. Musical sound signals analysis/synthesis: Sinusoidal+residual and elementary waveform models. *Applied Signal Processing*, 4:131–141, 1997.