RECENT IMPROVEMENTS OF AN AUDITORY MODEL BASED FRONT-END FOR THE TRANSCRIPTION OF VOCAL QUERIES

T.De Mulder[1]*, J.P.Martens* [1]*, M.Lesaffre*[2]*, M.Leman*[3]*, B.De Baets*[3] *and H.De Meyer*[4]

[1] Department of Electronics and Information Systems (ELIS), Ghent University;
Sint-Pietersnieuwstraat 41, 9000 Gent (Belgium). {tdmulder,martens}@elis.ugent.ac.be
[2] Institute for Psychoacoustics and Electronic Music (IPEM), Ghent University
[3] Department of Applied Mathematics, Biometrics and Process Control, Ghent University
[4] Department of Applied Mathematics and Computer Science, Ghent University

ABSTRACT

In this paper recent improvements of an existing acoustic front-end for the transcription of vocal (hummed, sung) musical queries is presented. Thanks to the addition of a new second pitch extractor and the introduction of a novel multi-stage segmentation algorithm, the application domain of the front-end could be extended to whistled queries, and on top of that, the performance on the other two query types could be improved. Experiments have shown that the new system can transcribe vocal queries with an accuracy ranging from 76 % (whistling) to 85 % (humming), and that it clearly outperforms other state-of-the art systems on all three query types.

1. INTRODUCTION

In the future, music consumers will have access to larger and larger music collections. Consequently, they will need efficient tools for the retrieval of specific musical material. Many people are already using electronic search and retrieval tools, tools that support textual specifications of title, performer, composer, date, label, etc. Research in music content analysis aims at the development of complementary and possibly more natural query methods. One such a method is Query-by-Melody (QBM) (see [1]) in which the user hums, sings, whistles or plays (on an instrument) a passage of the melody he wants to retrieve.

All existing QBM systems (e.g. [2, 3, 4]) seem to consist of two parts: (i) an acoustic front-end to transcribe the acoustic input into a sequence of note segments with their associated note frequencies, and (ii) a pattern matching back-end to search for the musical piece that best matches the provided transcription. In this paper, the focus is on the acoustic front-end and its transcription performance. The impact of this performance on the music retrieval quality of a QBM system is currently being investigated.

In [5] we already proposed a first auditory model based acoustic front-end that could transcribe singing sequences. In this paper we present important improvements of that front-end. We first present the novel algorithms that were developed (section 2), then we briefly describe the benchmarking tests we conducted (section 3), and we end (section 4) with a review of the experimental results we obtained with three acoustic front-ends.

This research was funded by the Flemish Institute for the Promotion of Scientifi c and Technical Research in Industry (project "Musical Audio Mining", 010035-GBOU). The authors acknowledge P.Y. Rolland, G. Raskinis and T. Heinz for granting permission to publish our results obtained with the Solo Explorer and Ear Analyzer.

2. THE NEW ACOUSTIC FRONT-END

The acoustic front-end proposed in [5] incorporates an auditory model (see [6]) and a note segmentation module. The auditory model consists of a cochlear processor and some additional modules for modeling primary aspects of central auditory processing (e.g. pitch and auditory spectrum extraction). The cochlear processor comprises a number of parallel channels each consisting of (1) a critical band wide band-pass filter tuned to a particular *channel frequency*, (2) a forward-driven automatic gain controller, and (3) a temporal envelope extractor. The channel outputs represent auditory nerve patterns that are partially synchronized with the input signal (up to 300 to 500 Hz). Consecutive channels have frequencies that are equidistantly spaced on a subjective frequency scale. The unit of that scale is the bark [7], and the frequency of channel m is equal to u_{cm} on this scale.

The pitch extractor (AMPEX) of the original model performs a temporal analysis of the high-pass filtered auditory nerve patterns but is incapable of detecting the pitch of e.g. a whistled sound. Therefore, another pitch extractor (SHS) is added to the model (Figure 1). It performs an analysis of the auditory spectrum $\mathbf{Y}$ derived from the low-pass filtered auditory nerve patterns. This analysis is inspired by the Sub-Harmonic Summation Theory of Terhardt et al [8]. The new auditory model thus generates an auditory spectrum and two pitch + voicing combinations per frame.

2.1. The new SHS pitch extractor

If the pitch of a periodic signal is sufficiently low, the critical band filters of most auditory channels capture several harmonics that interact and evoke periodic envelope patterns. However, if the pitch of the input signal becomes higher, most channels capture only one harmonic and exhibit no periodic envelope patterns anymore. On the other hand, consecutive harmonics appearing in different channels give rise to distinct maxima in the auditory spectrum, and the pitch can emerge from the positions of these maxima. This calls for a pitch extractor working in the frequency domain (analysis of the auditory spectrum). Our SHS algorithm therefor starts by searching for maxima in the auditory spectrum $\mathbf{Y}(n)$ of frame n. Then it subjects a maximum at position m^* to the following analysis:

Maximum position refinement. Define the region (m^*-2 .. m^*+1) or (m^*-1 .. m^*+2) (depending on whether $Y_{m^*-1}(n)$ is either larger or smaller than $Y_{m^*+1}(n)$), and check whether all $Y_m(n)$

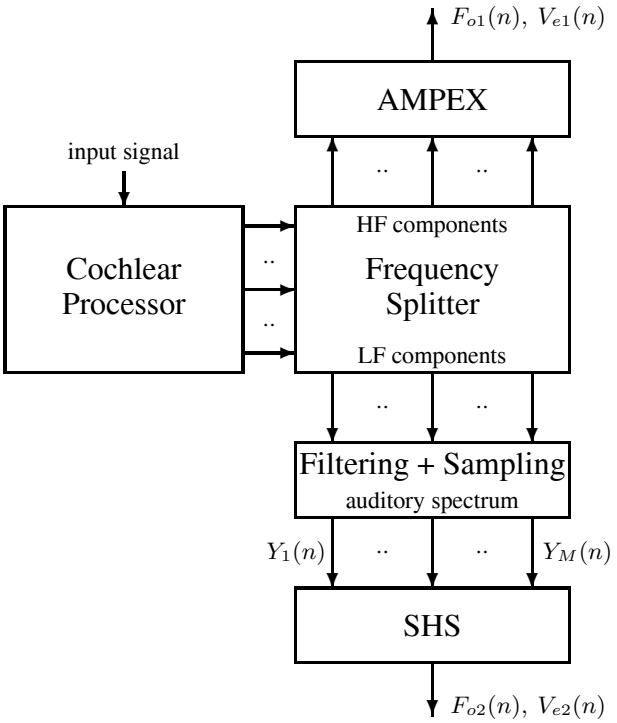

Fig. 1. Architecture of the newly presented auditory model.

in that region are smaller than $Y_{m^*}(n)$. If so, a parabola is fit to $Y_m(n)$ in the defined region and the position (u^*) and value (A^*) of its maximum are determined.

Maximum acceptance test. Compare $Y_m(n)$ with the auditory spectrum $T(u;u^*,A*)$ that would emerge from a pure tone of frequency u^* and amplitude A^*, and which is approximated by

$$\begin{aligned} T(u;u^*,A^*) &= A^*\left[(1-u+u^*)e^{u-u^*}\right]^{0.60} \quad u<u^* \\ &= A^*\left[(1+u-u^*)e^{u^*-u}\right]^{0.35} \quad u\geq u^* \end{aligned}$$

If $Y_m(n) \leq 1.05\, T(u_{cm};u^*,A^*)$ in an area of more than 2 bark wide around u^*, then a pure tone component of amplitude A^* and frequency F^* (in Hz) (corresponding to u^* in bark) is identified.

Pitch candidate detection. For each frame n one has obtained a tone set $\{F_k(n), A_k(n);\ k=1,..,K_n\}$. If it is not empty, the five sub-harmonics $F_k(n)/i\ \ (i=1..5)$ of each tone $F_k(n)$ are considered as potential pitch candidates, provided they fall in the range from 350 to 4000 Hz.

Pitch evidence computation. For each identified candidate $F_o(n)$, make a weighted sum of the amplitudes of the pure tones $F_k(n+j)$ in three frames ($j=-1..1$) that *coincide* with one of the harmonics of $F_o(n)$: F_1 and F_2 are said to coincide if $|F_1-F_2|/|F_1+F_2|$ is smaller than some ϵ_F. Introducing $\text{coinc}(F,F_o)=1$ in case of coincidence and 0 otherwise, we obtain

$$E_\gamma(F_o(n)) = \sum_{j,k} \gamma^i A_k(n+j)\ \text{coinc}(F_k(n+j),F_o(n))$$

The weighting with γ^i ($0<\gamma<1$) must prevent that sub-harmonics of the true F_o are selected as the pitch.

Pitch refinement. Once the pitch candidate $F_o(n)$ with the highest evidence is determined, its frequency is further refined to

$$F_o^*(n) = \frac{\sum_{j,k} \frac{F_k(n+j)}{i_{kj}} A_k(n+j)\ \text{coinc}(F_k(n+j),F_o(n))}{E_1(F_o(n))}$$

with i_{kj} being the harmonic ratio for which coincidence was established. I.e., if $F_k(n+j)$ coincides with $i_{kj}F_o(n)$, it attributes evidence for a pitch $F_k(n+j)/i_{kj}$. Here, there is no need for γ-weighting anymore as there is no competition between $F_o(n)$ and any of its sub-harmonics.

Evidence refinement. Once the refined pitch $F_o^*(n)$ is computed, its final evidence is computed as $E_1(F_o^*(n))$.

The free parameters of the SHS algorithm are γ and ϵ_F. They will be optimized experimentally. As SHS is only intended for pitches >400 Hz, it can achieve a sufficiently high resolution with a distance of 0.5 bark between channel frequencies.

2.2. The new note segmentation module

In [5] the segmentation of a query into note segments and rests was mainly based on an analysis of the total energy pattern $E(n)$ (= sum over channels of $Y_m(n)$). Now we propose a multi-stage algorithm that can cope better with legato, vibrato and tremelo. The different stages can be described as follows:

Stage 1: pre-segmentation. In this stage, candidate note boundaries are generated at clear minima in $E(n)$ and at places where $E(n)$ drops below a rest threshold for several consecutive frames (see [5]). Each segment between consecutive boundaries is labeled as rest (R) or note (N).

Stage 2: segment labeling. Every note is relabeled as rest (R), low-frequency note (LF) or high-frequency note (HF). To that end, two segment features are computed:

$$v_1 = \frac{\max V_{e1}}{V_{1,min}},\ \ v_2 = \frac{\max V_{e2}}{V_{2,min}}$$

with $\max V_{e1}$ and $\max V_{e2}$ being the maximum AMPEX and SHS evidences found in the segment. If $v_1 \geq 1$ then the segemt is marked as an LF note, else, if $v_2 \geq 1$ it is marked as an HF note, else is marked as R (rest). For each LF and HF segment we then determine a segmental pitch (see [5]) using either the frame pitches of AMPEX (for LF) or SHS (for HF).

Stage 3: boundary elimination. If t_{bound} is the boundary position, and E_{left} and E_{right} the energy maxima in the preceding and succeeding segment, the depth of the energy dip is defined as

$$\text{depth} = \frac{E(t_{bound})}{\min[E_{left},E_{right}]}$$

and a dip is considered *weak* if its depth exceeds some threshold ϵ_{depth}. A weak boundary is eliminated on the basis of its depth and the difference ΔF_o (in semitones) between the segmental pitches in the two surrounding segments:

$$\text{eliminate boundary if}\ \ \Delta F_o < a\ \text{depth} + d$$

Experiments revealed that it is better to adopt different combinations, (a_{LF},d_{LF}) and (a_{HF},d_{HF}), for LF and HF segment initial boundaries respectively.

Stage 4: legato processing. Not all boundaries between notes are marked by an energy dip. In order to retrieve these boundaries too, each generated long note segment (> 300 ms) is subjected to the following analysis:

- **Pitch stability analysis**. Determine for each frame the maximum interval to the right in which the minimum and maximum pitch still coincide (as defined before). The result of this analysis is a stable interval length pattern.
- **Stable interval detection**. From left to right, search for a maximum in the stable interval length pattern, and if it exceeds 150 ms, mark the interval starting at that maximum as a stable pitch interval and move to the position right after that interval. Repeat this procedure on the remainder of the segment until the end of the segment is reached.
- **Legato decision**. If there are multiple stable intervals, then consider the centers of the gaps between these intervals as boundaries and compute the segmental pitches of the newly created note segments.

The free parameters of the algorithm are $V_{1,min}$, $V_{2,min}$, ϵ_{depth}, (a_{LF}, d_{LF}) and (a_{HF}, d_{HF}).

3. FRONT-END EVALUATION

In order to evaluate the quality of an acoustic front-end one needs (i) a representative set of vocal queries and their correct transcriptions, and (ii) a good method for measuring discrepancies between the automatically generated and the correct transcriptions[1].

3.1. A manually labeled query database

For the evaluations presented in the next section, we have used 52 queries from 43 subjects. All queries were manually segmented in notes and rests, and a frequency (in Hz) was assigned to each note (see [9] for more details). The queries were collected in two campaigns: 18 queries (the ones also used in [5]) of 9 subjects were recorded in a quiet room, 34 queries of 34 subjects were recorded in a computer room with noise. The latter actually make part of a large set of 1148 queries of 79 subjects (see [9]). All queries were sampled at a rate of 22.05 kHz and amplitude normalized (same maximum for all queries). The queries were divided in four data sets:

1. **Develop**: 4 singing sequences, 2 whistled sequences and 1 singing+whistling sequence (234 notes).
2. **Syllables**: 19 syllable sequences (414 notes, 15 subjects).
3. **Words**: 19 word sequences (657 notes, 17 subjects).
4. **Whistled**: 7 whistled sequences (283 notes, 4 subjects).

The first set can be used for algorithm development, the others for testing. The test sets do not contain queries of subjects that appear in the development set.

While singing with syllables, subjects used different types of syllables (e.g. /la/, /di/, /na/). While singing with words, subjects did not necessarily use the words appearing in the artist performance, and often, they also used syllables at some instances.

[1]Both the queries and the evaluation software are available on http://www.ipem.ugent.be/MAMI.

3.2. A robust alignment procedure

As in [5], we align the automatically generated transcription with the correct transcription and we derive discrepancies from that alignment. The DTW-procedure is described in [5] but altered in three respects:

1. Rests are removed from the transcriptions before supplying them to the aligner.
2. The original timing cost c_{time} for assigning generated segment $(t_{g,i}, t_{g,i+1})$ to correct segment $(t_{c,j}, t_{c,j+1})$ is replaced by a segment overlap cost

$$c_{overlap} = 1 - \frac{\min(t_{g,i+1}, t_{c,j+1}) - \max(t_{g,i}, t_{c,j})}{t_{g,i+1} - t_{g,i}}$$

   It the segments do not overlap in time, $c_{overlap} > 1$.
3. The substitution cost is weighted with the fraction of the generated segment that overlaps with the associated correct segment (0 in case of no overlap). This way, short inserted notes are aligned mainly on the basis of their positions, and not so much on the basis of their (often wrong) frequencies.

Since the note onsets of some acoustic front-ends appear to be shifted in time with respect to the correct note onsets, an optimal time shift is superimposed on the generated onsets. The evaluation software automatically determines an optimal shift per file (in the range -2s to +2s) and keeps the corresponding alignment.

4. EXPERIMENTAL RESULTS

The main goal of our experiments is assess the differences between the transcriptions generated by the tested front-ends and the corresponding correct transcriptions provided with the tested queries[2].

4.1. Considered front ends

The front-ends investigated here are: (1) **Solo Explorer**, developed at the Information Processing Lab of the University of Paris [10] and now commercialized by Recognisoft; (2) **Ear Analyzer**, a physiological ear model developed by Heinz & Brückmann [11] at the Fraunhofer Institute in Ilmenau (Germany) and (3) **MAMI**, the system described in this paper (40 channels with channel frequencies ranging from 140 to 9000 Hz).

4.2. Determination of free parameters

The Solo Explorer and Ear Analyzer front-ends did not require any free parameter settings. For the MAMI front-end, the free parameter were fixed on the basis of tests performed on the development data. For SHS we found $\gamma = 0.75$, $\epsilon_F = 0.025$; for segment labeling: $V_{1,min} = 0.375$ and $V_{2,min} = 0.325$ times the maximal AMPEX and SHS evidences encountered in the development data; for boundary elimination: $\epsilon_{depth} = 0.3$, $(a_{LF}, d_{LF}) = (3, -1.5)$ and $(a_{HF}, d_{HF}) = (2, 0)$.

The most critical parameters are $V_{1,min}$, $V_{2,min}$ and the (a, d) sets used in boundary elimination. However, there is a broad area around the chosen settings where the performance remains stable.

[2]We have evidence that there is a strong correlation between the error measures computed for the front-end and the music retrieval error rates obtained with the outputs of that front-end in a Query-by-Melody system.

4.3. Evaluation of different front-ends

We tested all the acoustic front-ends on the three test sets. The measured errors (see Table 1) are deleted and inserted notes (= segmentation errors), and notes whose MIDI-code differs more than 1 from that of the corresponding correct note (=frequency errors). The MAMI front-end outperforms the other front-ends on all three test sets, be it that not all differences are statistically significant. The MAMI front-end performs especially better on the transcription of word sequences. Its transcriptions of whistled sequences are not significantly better than those emerging from the Solo Explorer. For some files, Ear Analyzer produces just a few notes (more than 80% note deletions), whereas for others it yields a very reasonable performance.

test set	error type	Evaluated acoustic front-ends Solo	EarAn	MAMI
syllables	del+ins	20.1	15.9	10.4
	ΔF	2.9	4.6	4.1
	total	23.0 ± 4.0	20.5 ± 3.9	15.5 ± 3.5
words	del+ins	24.3	48.5	15.4
	ΔF	8.7	13.3	5.8
	total	33.0 ± 3.6	61.8 ± 3.7	21.2 ± 3.1
whistled	del+ins	24.7	35.0	20.8
	ΔF	4.2	2.5	2.8
	total	28.9 ± 5.3	37.5 ± 5.6	23.6 ± 4.9
all data	del+ins	23.1	35.2	15.0
	ΔF	5.9	8.4	4.7
	total	29.0 ± 2.4	43.6 ± 2.6	19.7 ± 2.1

Table 1. Evaluation of three front-ends. Listed are percent segmentation errors (del + ins) and frequency errors (ΔF). Total error rates are given with their 95% confidence intervals.

4.4. Evaluation of different MAMI versions

In a second experiment we have compared different versions of the MAMI front-end. The versions are annotated as Mxy with $x = 1$ if boundary elimination was used and $y = 1$ if legato processing was used. The performances of the different versions are summarized in Table 2.

data set	error type	Evaluated MAMI front-ends M00	M10	M01	M11
syllables	del+ins	10.6	9.9	12.6	10.4
	Δ	5.1	4.6	4.1	4.1
words	del+ins	17.4	16.6	16.6	15.4
	Δf	5.5	5.5	5.8	5.8
whistled	del+ins	59.7	21.9	59.0	20.8
	Δf	4.2	2.8	4.2	2.8

Table 2. Comparison of the four MAMI front-ends mentioned in the text

The first apparent conclusion is that the boundary elimination can avoid a lot of note insertions during whistling without increasing the number of deletions in other types of notes.

The second conclusion is that legato processing is not useful when it is applied directly after the pre-segmentation (as in M01). It may even degrade the performance, as e.g. for syllable sequences. However, it does yield some (non-statistically significant) improvement when applied in combination with the full-scale segmentation algorithm. We observed that in the development data, the legato processor could find most of the legato's without introducing note insertions. In the test data however, 3 out of 10 detected legato's appeared to be false alarms.

5. CONCLUSIONS

The main conclusions of our work are that the newly presented acoustic front-end can transcribe all types of vocal queries: its accuracy (100% - error rate) ranges from 76 % for whistled queries to 85% for syllable sequences. The newly presented front-end outperforms the other tested state-of-the-art systems on all data sets. It is also established that most of the errors are segmentation errors. It is our feeling that in the future one should continue to improve the query segmentation strategy.

6. REFERENCES

[1] Shwartz S., Dubnov S., Friedman N., and Singer Y., "Robust temporal and spectral modeling for query by melody," in *Procs. SIGIR*, 2002, pp. 331–338.

[2] McNab R., "The new zealand digital library melody index," in *D-lib Magazine*, 1997, pp. 11–18.

[3] Kosugi N., Nishihara Y., Sakata T., Yamamuro M., and Kushima K., "A practical query-by-humming system for a large music database," in *Procs. ACM Multimedia*, 2000, pp. 333–342.

[4] Pauws S., "Cubyhum: A fully operational query by humming system," in *Procs. ISMIR*, Paris, 2002, pp. 187–196.

[5] Clarisse L., Martens J.P., Lesaffre M., De Baets B., De Meyer H., and Leman M., "An auditory model based transcriber of singing sequences," in *Procs. ISMIR*, Paris, 2002, pp. 116–123.

[6] Van Immerseel L. and Martens J.P., "Pitch and voiced/unvoiced determination with an auditory model," in *J. Acoust. Soc. Am. 91*, 1992, pp. 3511–3526.

[7] Zwicker E. and Feldtkeller R., Eds., *Das Ohr als Nachrichten Empfanger.*, Hirzel Verlag Stuttgart, 1967.

[8] Terhardt E., Stoll G., and Seewann M., "Algorithm for extraction of pitch and pitch salience for complex tonal signals," in *J. Acoust. Soc. Am. 71*, 1982, pp. 679–688.

[9] Lesaffre M., Moelants D., and Leman M., "Spontaneous behavior in vocal queries for audio mining," in *Procs.*, 2003, pp. 333–342.

[10] Rolland P.Y., Raskinis G., and Ganascia J., "Musical content-based retrieval: an overview of the melodiscov approach and system," in *Procs. ACM Multimedia*, 1999, pp. 81–84.

[11] Heinz T. and Brückmann A., "Using a physiological ear model for automatic melody transcription and sound source recognition," in *Procs. 114th AES Convention*, Amsterdam, 2003.