Musical Content-Based Retrieval : an Overview of the Melodiscov Approach and System

Pierre-Yves Rolland
LIP6
CNRS-UPMC
4 place Jussieu, 75005 Paris
Tel.: +33 1 4427 8800
P_Y_Rolland@yahoo.com

Gailius Raškinis
LIP6 and Vytautas Magnus University
Vileikos 8
3035 Kaunas, Lithuania
Tel.: +370 7 796 792
idgara@vaidila.vdu.lt

Jean-Gabriel Ganascia
LIP6
CNRS-UPMC
4 place Jussieu, 75005 Paris
Tel.: +33 1 4427 3727
ganascia@apa.lip6.fr

Keywords

Music, Content-based information retrieval, *'WYHIWYG'*, automated music transcription, multimedia knowledge representation, pattern matching, machine learning.

1. INTRODUCTION

Research and development related to content-based retrieval of music/audio has been receiving increasing attention. Its application potential is tremendous for any musical information retrieval context, including online purchase of music. This includes buying traditional audio CDs as well as, in a more recent paradigm, purchasing single songs/pieces by downloading them in some compressed format. This information retrieval paradigm consists of allowing the user to retrieve musical/audio material based on a query that is either hummed/sung or played on a MIDI keyboard (actual or virtual). We concentrate here on the first setting — the most practical and difficult one — which we propose to dub WYHIWYG (What You Hum Is What You Get). The input of the process is thus a digital audio signal resulting from the sampling of the users' vocal audio signal. The ouput is a list of pointers to music resources. Such a pointer may be a hyperlink to an online form for purchasing the musical piece or the CD containing it, or a hyperlink to a Midifile freely available on the WWW.

Melodiscov - which stands for Melody Discovery - is a novel WYHIWYG system we have designed and implemented (prototype). The main two processing stages in Melodiscov are named transcription and search. Transcription transforms sampled acoustic data into an intermediate form suitable for subsequent search(es) of the target material, which we refer to as the database. Transcription applies both to the sung/hummed query, and to each of the database's musical pieces when these are initially available in audio form (as opposed to, e.g., MIDI form). In this paper we will first describe in some detail Melodiscov's transcription component. Key aspects of its search component will then be presented. A critical discussion of Melodiscov's characteristics, enriched by preliminary experiments/tests with the system, will follow.

2. TRANSCRIPTION

Transcription is taken here to designate the process of converting an acoustic input into a 'symbolic' musical representation such as MIDI-type coding or standard music notation. The process of monophonic (monodic) transcription can be presented as a chain of several information processing steps: digital signal processing, segmentation, and quantization of segment duration and pitch (Fig. 1). The goal of signal processing is to extract temporal sequences (tracks) of useful features. The most widely used features are amplitude/power and pitch. The goal of the segmentation procedure is to divide an acoustic stream into notes. Quantization procedures are responsible for labeling each note with a discrete note name and a rhythmic value.

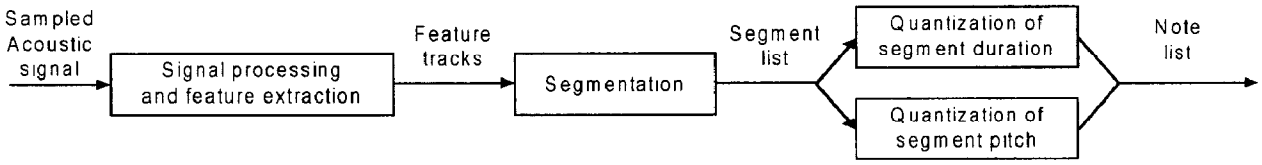

Fig.1. Monophonic (monodic) transcription scheme.

As far as we know, the transcription of vocal music for melody retrieval was attempted by Ghias et al. at Cornell University [2], McNab et al. at the University of Waikato [3],[4] and Pollastri at the University of Milano [6]. One key limitation identified in the corresponding transcription components is the constraints they place on the acoustic input, e.g. imposing the user to sing/hum separate notes and/or use percussive syllables such as "ta-ta" or "da-da". Another is their strong sensitivity to the presence of vibrato and to out of tune singing. The transcription component we developed for Melodiscov is a contribution in overcoming these limitations (see section 5 for more discussion on this). This component, which follows the general scheme of Fig. 1., has been dubbed ATRAMA for Automatic TRAnscription of Monophonic Audio. (In Lithuanian *atrama* means 'support'). Originally, it was designed for the difficult problem of automatically transcribing strongly ornamented vocal signals such Lithuanian folk songs [7]. The algorithm breaks the acoustic signal into 33-40 ms overlapping frames and extracts a RMS-power (p) and a pitch (π) estimate for each frame. Pitch extraction is based on a cross-correlation technique similar to the one described by [5]. The choice of correlation techniques was motivated by two reasons: they are robust and can make pitch estimates as accurate as desired by means of interpolation. Instead of taking pitch decision based on current frame's content, our algorithm stores the set of prominent peaks of the cross-correlation function. These peaks represent pitch candidates for that frame. The final decision is delayed until candidate sets have been established for all frames. Thus, pitch decision is made more robust as it explores the full set of constraints imposed on pitch continuity. Additional features obtained during the step of signal processing include power and pitch derivatives (dp, $d\pi$) and their relative derivatives ($\delta p = dp/p$, $\delta\pi = d\pi/\pi$).

Segmentation is preceded by an automatic vibrato detection phase. Vibrato is detected by searching the pitch track for regularly oscillating patterns. The segmentation algorithm integrates both power-based and pitch-based features. Easier segmentation cases are derived by means of empirically stated rules based on the combination of the aforementioned features. The power p is examined for silence threshold crossovers; δp is inspected for cumulative increase/decrease in power, the integration being based on multiple time intervals of different length. The pitch track π is examined for significant frequency changes and for the succession of voiced and unvoiced frames. Difficult segmentation cases or "suspicious" cases (Fig. 2) are stated if a sound fragment matches segmentation rules partially. We distinguish several types of "suspicious" cases. The approach used for making decisions in difficult cases relies on machine learning techniques. This means that decision rules are automatically learnt from examples based of cases pairs, individually.

ATRAMA makes no assumptions about the rhythmic structure of the musical content under investigation, but assumes locally constant tempo. The module of rhythm quantization uses histogram methods for clustering note durations. Duration clusters provide a starting point for an iterative relaxation algorithm. This algorithm brings local corrections to individual note durations until they converge or fail to converge towards the multiples of a basic duration unit. For the purposes of WYHIWYG applications, the quantization step may optionally be omitted.

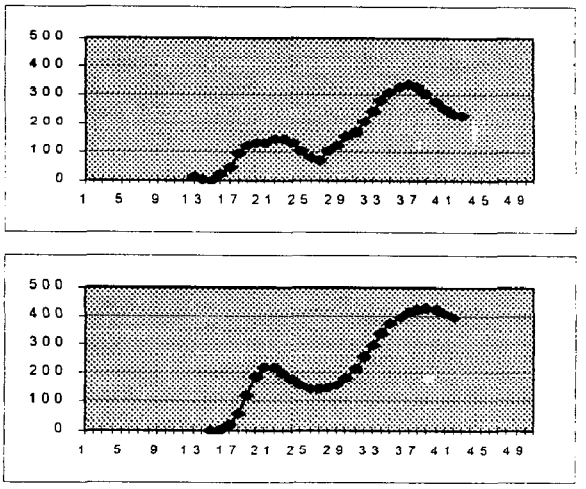

Fig. 2. The illustration of a difficult segmentation case. Pitch contours of both sound fragments seem similar.
***Top:* two notes performed in vibrato style.**
***Bottom:* one note with glissando (portamento) preceding it.**
(1 horizontal graduation = 10 ms; vertical axis represents pitch in cents)

The pitch quantization module assumes that singer's tuning may be imperfect. It is designed to deal with reasonably compressed or stretched melodic scales and with possible evolutions of this scale over time. The pitch quantization module realizes two tasks. First, it assigns a single continuous pitch "representative" to each segment. Second, it maps this representative to MIDI note number. Almost all known transcription systems use pitch average as a representative of segment pitch. Our algorithm considers this as the solution of last resort, when sustained parts or regular vibrato are not present inside of a given segment. Continuous to discrete value mapping is realized through three steps. First, the relationship (constraint) graph is constructed where constraints encode both near and distant note-to-note relationships. Second, constraint propagation techniques are used to explore this graph. This step results in a number of disjoint subsets of internally quantized notes. Third, note subsets are unified. Here, the algorithm uses heuristics inspired by music theory. For example, if two concurrent quantization are possible, the one that best fits the diatonic scale is preferred.

3. SEARCH

The second stage in Melodiscov's processing scheme, search, in turn divides into two sub-stages. One involves an enrichment or change of the representation of music, as explained below. The other is named pattern matching. Here, pattern matching refers to the process of looking for occurrences of a sequential pattern over a given alphabet within a set of sequences over the same alphabet. The sequential pattern is the result of automatically transcribing the hummed/sung query while the sequence set is the music database. This is a classically solved string processing problem, which won't be detailed here. The major point is that every algorithm of this kind relies on an

explicit or explicit sequential similarity model [10]. In turn, every such model relies on a particular representation of sequences and their elements (viz. music pieces/songs and notes). The similarity model provides a quantitative measure (or, in certain cases, a qualitative one) of any pair of segments being compared. The proposition P(Pat,Pos,Seq), "having an occurrence of the query pattern Pat at position Pos of sequence Seq in the database" is defined using the model:

> P(Pat,Pos,Seq) is TRUE iff there exist a segment (or factor) Seg of Seq, starting at position Pos, such that the computed similarity between Seg and Pat is above a given threshold T.

Previous research and development into musical content-based retrieval has used very simple representations of music such as note properties of absolute pitch or pitch interval size or direction, and durations or duration ratios. Similarity models used are also simplistic, namely basic edit distance-based ones (for efficiency optimization purposes, among others—see section 5). We claim that significant limitations of these systems are due to the straightforward character of these representations and similarity models. We base this claim on work that has been made in the areas of psychology of musical perception and cognition on one hand, and in computer science/artificial intelligence and music on the other (See e.g. [1],[10]).

First, appropriately modeling musical similarity requires the use of musical descriptions that match those identified by psychology and music theory at large. These are not limited to pitch intervals—identified as playing a key role in perceiving or remembering melodies. In Melodiscov we designed and implemented a representation enrichment/change process. This process is carried out on the database prior to searching, and on each new query as a preprocessing sub-stage within search. It results in a structured representation of the query and database pieces/songs that uses object-oriented data structures. In Melodiscov, individual, local and global descriptions can be automatically computed from the initial representation of music (viz. MIDI-type encoding in the case of the transcribed query). An individual description (e.g. pitch, duration, intervals) concerns a particular element (note/rest). A local description (e.g. ascending contours or "gap-fills") concerns a particular segment (passage). A global description (e.g. average pitch or duration, major/minor mode) concerns a particular sequence (melody).

Second, appropriately modeling musical similarity requires gradual sequential comparison paradigms which take into account all relevant/chosen characteristics of notes and pieces/songs. We designed a model dubbed MVEM (multidescription valued edit model [8] and subsequently implemented it within Melodiscov. This model allows to simultaneously take into account any set of individual, local or global descriptions (stemming from the representation enrichment/change preprocessing). Each description receives a specific weight (defaulting to 1). Descriptions and weights are taken into account in computing the contribution of every allowed edition to the overall similarity. Allowed editions include insertions (of one or several notes), deletions (of one or several notes), substitutions (of one note by another), consolidations, fragmentations, swaps and generalized substitutions. The contribution of a "drastic" edition (e.g. the deletion of a very salient note: particularly long duration, atypically high pitch, etc.) will be a very negative number. Conversely, the contribution a "mild" edition (e.g.: replacement of a note by a note very similar to it for all descriptions) will be a moderately negative, or even a positive, number. By contrast, in most approaches, including previous WYHIWIG systems, insertion, deletions and substitutions receive a constant cost, regardless of the various musical characteristics involved.

The user is placed at the core of both the representation enrichment/change process and the MVEM. Melodiscov provides the user with the possibility to both choose what descriptions should be used and what respective weights they should have in melodic comparison. This materializes the adoption of different viewpoints on melodic similarity and, hence, on search results. For instance, the user may at some point privilege temporal descriptions (durations, metrics, etc.) w.r.t. frequential descriptions (pitches, intervals, etc.) to achieve more rhythmically-oriented searching.

4. QUALITATIVE RESULTS

Preliminary experiments have been conducted using a (small) database of pop songs downloaded as publicly available Midifiles from WWW sites. In each test, one of the subjects (most without any musical training) was requested to sing/hum a fragment of one of the songs he knew. Melodiscov then searched the database and returned best matches. This was repeated several times for each subject using a different song and/or a different input mode. This means that subjects alternatively used humming (mm-mm or aa-aa), "percussive" singing (da-da or ta-ta —different vowels were investigated) and singing with lyrics. Singing skills/ ability ranged from very poor to excellent. Of course, subjects were not allowed to listen to the songs just before making tests. These audio queries were used with another online (WWW) melodic content-based retrieval system and with the audio-to-midi component of a widely used software sequencer, for comparison purposes. Obviously that comparison only concerned transcription aspects, as the databases were different.

We obtained very satisfactory results overall, which went beyond our expectations. More than half of the queries were transcribed with over 90% accuracy in absolute pitch (MIDI note number). This included particularly 'difficult' queries: 17 notes long, only small (ambiguous) intervals, complex rhythm. In terms of intervalic direction or even interval size (small/large), among all the queries hardly any was transcribed with less than 100% accuracy. Concerning rhythmic aspects, ATRAMA's performance was favorably evaluated through its capacity to correctly delimitate start- and end-times of transcribed notes. Even in the most difficult situation (inaccurate singer + lyrics) results were very good. Comparisons with the other system gave a very clear-cut advantage to Melodiscov's ATRAMA. In a very illustrative example, the query consisted of 11 notes of the main part of the chorus in the Beatles' song Penny Lane, sung with lyrics. 10 out of the 11 notes were correctly transcribed by Melodiscov (absolute pitches and durations). With the online system only 3 notes (the longest ones in the query) were transcribed, and with very excessive durations.

Additionally, tests were made where users made queries by whistling. Results obtained with Melodiscov were good and contrasted even more clearly with those of the other two systems using the same queries. It can be noted, though, that the online WYHIWYG system may not have been designed to handle such queries.

Of course full-scale testing will be needed to quantitatively confirm these results (see below).

5. DISCUSSION

Based on these preliminary results it can be said that Melodiscov yields satisfactory transcription performance without imposing any constraint on the user's way of singing/humming. In addition to whistled queries, as mentioned above, Melodiscov has also appeared to perfectly transcribe queries played on various musical instruments. The implementation (C language) of ATRAMA is not optimized for speed but gives satisfactory processing time (typically less than half the duration of the hummed/sung query). Not all potentially useful spectral features are presently taken into account (a topic currently under investigation). It would also be useful to enrich the set of "suspicious" pattern types handled (see Fig. 2). Similarly more learning examples would be needed to better handle difficult segmentation decisions.

On the search side, Melodiscov allows the user to base the search on a (possibly personalized) rich set of musically-meaningful descriptions. This tremendously increases search sensitivity (returning all appropriate pieces) and, in certain cases, search accuracy (returning only appropriate pieces). Using a richer (more flexible) sequential similarity model with a richer representation of musical information, can result in an increase of the search stage's execution time. For really large databases, our proposed solution is to offer the user, as an additional option, to choose between strict and flexible search mode. The slower flexible mode is then reserved for cases when the faster, strict mode has failed.

6. ON-GOING AND FUTURE WORK

The next step is the full-scale testing of our concepts, algorithms and system, possibly using publicly available folksong databases and a MIDI database of about 200 jazz transcriptions which we developed and used in another project started earlier (see e.g. [8],[9],[10]). A full-scale implementation, and the integration of a WWW interface, are scheduled to follow. Among longer-term perspectives is the following. We plan to apply FlExPat, an automated pattern discovery algorithm we designed and implemented [9], to further speed up search. FlExPat automatically locates all significant repetitions — whether exact or approximate— within the various sequences of the database, as well as significantly similar segments between different sequences. For every group of resembling segments (and there are huge numbers of such groups in practice), it can be sufficient to compare the query pattern once to the group's representative than to compare it to all of the group's segments. In addition to segment groups, FlExPat automatically computes such representatives (prototypes). In conclusion, tremendous efficiency increase w.r.t. existing WYHIWYG algorithms could be obtained via such a (pre)processing of the database before searching.

7. REFERENCES

[1] Cambouropoulos, E., T. Crawford & C.S. Iliopoulos. (1999). Pattern Processing in Melodic Sequences: Challenges, Caveats and Prospects. In Cambouropoulos, E. and Rolland, P.Y. (editors) Proceedings of Focus Workshop on Pattern Processing in Music Analysis and Creation. Edinburgh, April 1999.

[2] Ghias, A., J. Logan, D. Chamberlin, B.C. Smith (1995). Query by humming - Musical information retrieval in an audio database, ACM Multimedia'95 - Electronic Proceedings (http://www.cs.cornell.edu/Info/people/ghias/publications/).

[3] McNab, R.J., L.A. Smith, I.H. Witten, C.L. Henderson and S.J. Cunningham (1996). Towards the digital music library: tune retrieval from acoustic input, Proceedings of ACM Digital Libraries'96, pp. 11-18.

[4] McNab, R.J., L.A. Smith, D. Bainbridge and I.H. Witten (1997). The New Zealand digital library melody index, D-Lib Magazine (http://www.dlib.org/dlib/may97/meldex).

[5] Medan, Y., E. Yair and D. Chazan (1991). Super resolution pitch determination of speech signals, IEEE ASSP 39(1), pp. 40-48.

[6] Pollastri, E. (1998). Melody-retrieval based on pitch-tracking and string-matching methods, Proceedings of the XIIth Colloquium on Musical Informatics.

[7] Raškinis, G. (1998). Preprocessing of Folk Song Acoustic Records for Transcription into Music Scores, Informatica 9(3), IMI, Vilnius.

[8] Rolland, P.Y. 1998. Découverte Automatique de Régularités dans les Séquences et Application à l'Analyse Musicale. Thèse de Doctorat en Informatique de l'Université Paris VI (Ph.D). July 1998.

[9] Rolland, P.Y. (1998). FlExPat : a Novel Algorithm for Musical Pattern Discovery. Proceedings of the 12th Colloquium on Musical Informatics (XII CIM). pp. 125-128. Co-sponsored by IEEE (CS/TCCGM). Italy.

[10] Rolland, P.Y., Ganascia, J.G. (1999). Musical Pattern Extraction and Similarity Assessment. In Miranda, E. (ed.). Readings in Music and Artificial Intelligence. Harwood Academic Publishers.