Music Indexing and Retrieval for Multimedia Digital Libraries

Nicola Orio

Department of Information Engineering – University of Padua
Via Gradenigo, 6/b – 35131 Padua – Italy
nicola.orio@dei.unipd.it

Abstract. This chapter addresses the problem of the retrieval of music documents from multimedia digital libraries. Some of the peculiarities of the music language are described, showing similarities and differences between indexing and retrieval of textual and music documents. After reviewing the main approaches to music retrieval, a novel methodology is presented, which combines an approximate matching approach with an indexing scheme. The methodology is based on the statistical modeling of musical lexical units with weighted transducers, which are automatically built from the melodic and rhythmic information of lexical units. An experimental evaluation of the methodology is presented, showing encouraging results.

Key words: music retrieval, indexing, approximate matching, weighted transducers

1 Introduction

Users of multimedia digital libraries may have different knowledge and expertise, which are related to the ability to describe their information needs precisely. This is particularly true for music language, where the level of music education may vary remarkably among users, who may range from casual listeners to performers and composers. Untrained users may not be able to use bibliographic values or take advantage of metadata when searching for music. For this reason, the access to music digital libraries should be content-based.

The main idea underlying content-based access and retrieval is that a document can be described by a set of features that are directly computed from its content. This approach is the basis for most of the methodologies for information retrieval, where the content of a textual document – e.g. its set words – is automatically processed and used for indexing and retrieval. Even if multimedia data requires specific methodologies for content extraction, the core information retrieval techniques developed for text may be extended to other media because the underlying models, which are based on statistics and probability theory, are likely to describe fundamental characteristics being shared by different media, languages, and application domains [23]. For the particular case of music language, already in 1996 McLane [30] stated that a challenging research topic would be the application of some standard principles of text information retrieval to music representation.

As is well known, textual information retrieval is based on the concept that *words*, which form a document, can be considered as good content descriptors. Following this idea, documents can be efficiently described using words as index terms, and retrieval can be performed using a measure of similarity between query and documents indexes. If we follow the hypothesis that this principle can be extended to indexing and retrieval of music documents, then ad-hoc algorithms have to be designed to produce musical *lexical units*, like words in textual documents, and compute the *similarity* between such units.

2 Background

Before introducing the approach to music retrieval, some background information needs to be provided. First of all, the limitations of music metadata for a retrieval task are presented to highlight the need for content based approaches. This introduces the main question regarding music, that is, which content is conveyed by a music work. The basic concepts of music retrieval are then introduced by drawing a parallel between the text and the music domains based on the application of well known techniques.

2.1 Music Metadata

Textual metadata have been used for centuries as a tool for the concise and effective description of document content [12]. The extension to other media, such as images and video, proved to be effective as well, because metadata allows for the content of a whole document to be summarized with a small set of keywords. A number of music digital libraries are accessible through the use of metadata, such as Cantate [6] and Musica [36], which allow users access to choral music using metadata and lyrics.

Music metadata describes a number of characteristics of music documents, which can be divided in three main categories:

- *Bibliographic values*, which are shared by almost all media and, in the case of music, also include cataloguing number, the title of a complete work and the titles of its parts (or movements), the names of the authors and of the performers.
- *Information on music form*, which is typical of music, and gives information about genre, time and key signatures, tempo, orchestration, and so on.
- *Additional available information*, such as lyrics for vocal works and, if applicable, links to external documents that create a context for the music work (e.g. a drama, a movie, a poem).

In order to carry out an effective search through music metadata, a user needs to have a good knowledge of the music domain, which may not be the case for casual users of a music digital library. Moreover, music metadata have a number of limitations. For instance, in the case of tonal Western music, it is typical to have titles such as “Fugue in G major” or “Suite”, which describe the music form rather than the content. Moreover, the title is often based on some music features for other music genres as well. For instance, a user needs to be aware of the difference between a “jig” and a “reel” in order to effectively use these metadata to retrieve Irish music, and between a “bossa” and a “blues” to retrieve jazz music.

General information is often too generic to be a good discriminator between different music works. For instance, the genre information groups together hundreds of thousands of different works. Moreover, in tonal music there are only 21 major and 21 minor different tonalities, and thousands of compositions can be labeled with the terms “cantata” or “concerto” in classical music, and with the same terms “up tempo” or “slow” in pop and rock. This kind of metadata can be useful to refine the description of a music information need, but it can hardly be used to completely define it. Moreover, a preliminary study on users information needs, presented in [25], showed that users are interested in retrieving songs by their specific content. Additional information in the form of lyrics, when present, can be particularly useful to describe an information need, yet in this case the retrieval of music documents becomes an application of textual information retrieval. Contextual information, such as the movie where a particular soundtrack was used, or the poem that inspired a particular composition, can be very helpful as well to describe a user information need, yet this kind of contextual information applies only to a small percentage of music documents.

What is normally missing in music metadata is a textual description of the document content other than its musical structure, which is a peculiar situation of the music language that is due to the fact that music is not aimed at describing something with a known semantic – like text, images, speech, video, or 3D models. It should be considered that music representation is intrinsically limited because it is aimed at giving directions to performers and not to describe high level characteristics [34]. Finally, it is not clear yet, among music theorists and musicologists, whether music can be considered as a language and whether it describes something other than itself. This is probably the main limitation of the use of metadata for music indexing, and it is the motivation for the increasing number of content-based approaches proposed in recent years.

2.2 The Dimensions of Music

Rhythm, melody, and harmony are different dimensions that capture distinctive features of a music document, thus music has an instrinsic multidimensional nature. These dimensions are conveyed explicitly by music scores and recognized easily by listeners of audio recordings, and are used extensively by music theorists and musicologists as tools to describe, analyze, and study music works. Another perceptually relevant music dimension is timbre, which is related to the quality of sounds and is conveyed only by audio recordings. Yet timbre is a multidimensional feature by itself, and can be described using a set of continuous parameters, such as spectral power energy or Mel-Cepstrum Coefficients, and by perceptually based features such as spectral centroid, roughness, and attack time. As stated in [24], timbre remains a difficult dimension to understand and represent, though studies have been carried out on the perception of timbre similarity [4]. Other dimensions are related to the structure, to the music forms, to the orchestrations. The discussion on content-based music indexing, however, will be limited to the ones that have a symbolic representation, namely rhythm, melody, and harmony – with a special focus on melody. For any chosen dimension, the indexing scheme has to be based on a suitable definition of the particular lexical units of the dimension and their representation. A taxonomy of the characteristics of music and their potential interest for users is reported in [26].

The representation of the melody can build upon traditional score representation, which is based on the drawing of a sequence of notes, each one with a given pitch and a duration relative to the tempo of the piece. This symbolic representation is particularly suitable for indexing, providing that the melodic lexical units are highlighted. This is a more difficult task also for musicians and music scholars; the results of a perceptual study on manual segmentation are presented in a following section. The representation of rhythm can be considered as a variation of melodic representation, where pitch information can be discarded or substituted with the information of the particular percussive instrument that plays each rhythmic element. Similarly, the indexing of the harmonic dimension can be based on common chord representation. In this case there are alternative representations, from figured bass to functional harmony and chord names. An overview of chord representations, aimed at their annotation, is presented in [16]. The segmentation of chords in their lexical units can be based on notions of harmony, including modulations, cadences, and the use of particular chord progressions in different music genres.

The analysis of different dimensions and their representation as building blocks of music documents may be of interest also for musicologists, composers and performers. To this end, it is interesting to cite Humdrum [22], which incorporates retrieval with a number of tools for the manipulation and analysis of music scores.

2.3 Application of Information Retrieval Concepts to Music

Textual information retrieval, which is normally addressed as information retrieval *tout-court*, has a long research tradition and thus is a natural choice for the investigation of how the main concepts underlying the different methodologies can be applied to the music domain. In particular, there are four main steps related to textual document indexing that are relevant to this discussion.

1. Lexical analysis.
2. Stopwords removal.
3. Stemming.
4. Terms weighting.

These steps reflect some considerations on the content-based description of textual documents that have a parallel in the music domain, and are discussed in detail in the following sections. Among the different steps, the last two are the most relevant for the methodology proposed in this chapter.

Lexical Analysis

The basic idea underlying lexical analysis is that *words* are the most relevant content descriptors of a textual document. Thus lexical analysis corresponds to document parsing for highlighting its individual words, which is almost straightforward for many languages, where there is a clear separations between words – blanks, commas, dots, and so on. It can be noted that for some languages, such as Chinese and Japanese, the compounding of ideograms in different words has to be inferred from the context and is a non trivial task.

As regards the parallel of lexical analysis to the music domain the first issue involves the choice of the dimensions to be used as content descriptors. This choice influences the approaches to the lexical analysis. For instance, if rhythm is used to index music documents, the attack time of the different notes has to be automatically detected and filtered, which is an easy task for symbolic documents and can be carried out with good results for documents in audio format too. On the other hand, if harmony is used to compute indexes, lexical analysis has to rely on complex techniques for the automatic extraction of chords from a polyphonic music document, which is still an error prone task especially in the case of audio documents, even though encouraging results have been obtained [15]. For simplicity, it is assumed that a sequence of features is already available, describing some high-level characteristics of a music documents, related to one or more of its dimensions. It is also assumed that the feature extraction may be affected by errors, which should be taken into account during the design of a music retrieval system.

Even after a sequence of features have been automatically extracted, music lexical analysis remains a difficult task. The reason is that music language lacks of explicit separators between candidate index terms for all of its dimensions – like the previous example of documents written in Chinese or Japanese. Melodic phrases, rhythmic patterns and harmonic progressions are not contoured by special signs or sounds that express the presence of a boundary between two lexical units. This is not surprising, because lexical units are not part of the common representation of music documents. Even if there is a wide consensus in considering music as a structured organization of different elements, and not just a pure sequence of acoustic elements, historically there has been no need to represent this aspect directly. Music is printed for musicians, who basically need the information to create a correct performance, and who could infer the presence of basic elements from the context. In order to overcome this problem, different approaches have been proposed in the literature for lexical analysis, considering musical patterns [19], main themes [31], or musical phrases [32].

The consistency between musicians in performing the lexical analysis of some monophonic scores was investigated in a perceptual study [33], showing that all the participants perceived the presence of lexical units, but they were consistent in choosing boundaries only when strong cues were present. The lexical analysis of music documents is still an open problem, both in terms of musicological analysis because alternative theories have been presented, and in terms of indexing and retrieval effectiveness. An experimental evaluation of the effectiveness of different approaches to melodic segmentation was presented in [38]. Figure 1 shows the opening bars of *Psyché* by Jean-Baptiste Lully, which have been segmented using three different segmentation approaches, based on perception, statistical modeling and discontinuity detection, respectively.

Stopwords Removal

The main concept behind stopwords removal is that a subset of the words extracted through lexical analysis may be discarded, eventually affecting efficiency and effectiveness in a positive way. Such words may be either the ones that have only a grammatical function, and thus do not express any semantic, or the ones that are almost uniformly distributed across the documents.

In the case of music, it is difficult to state whether or not a musical lexical unit has a meaning, because it is not clear if music itself is aimed (or even able) to convey any meaning at all. On the other hand, to define how much a

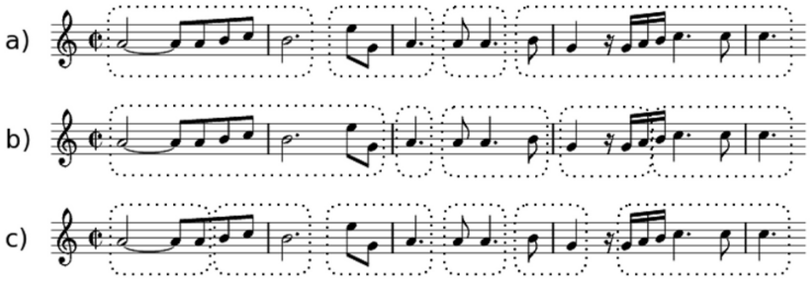

Fig. 1: Example of the results of different segmentation techniques based on: **a**) Gestalt principles, **b**) statistical modeling, **c**) discontinuity detection

particular unit is a good discriminator between different music documents is an easier task. For instance, a lexical unit of two notes, with identical length, that form a major second is likely to be present in almost all works of vocal music, and thus is probably a poor descriptor of music content. Depending on the particular set of features used as content descriptors, the designer of a music retrieval system can make a number of choices about the possible stop-lists of lexical units, which could be driven by both musicological and computational motivations and by the characteristics of the music collection highlighted by a statistical analysis. It should be noted that this approach is not usually exploited in the music information retrieval (MIR) literature, where the term stop-list is seldom used. The common approach is to carefully select the processing parameters to avoid the computation of lexical units that are believed to be uninformative.

Stemming

As described in a previous Chapter, the idea behind stemming is that two index terms may be different but can convey similar meanings. Analogously, two musical lexical units may be slightly different, yet listeners can perceive them as almost identical, or confuse one with the other when recalling them from memory, or consider that they play a similar role in the musical structure. Examples of potential applications of stemming are: identical rhythmic patterns played at a different tempo, melodies that differ only for few intervals that from major turn minor and viceversa, chord progressions where some chords are substituted by others with a similar function.

A way to take into account variants in lexical units is *quantization*. The main motivation of quantization in music processing is probably related to the fact that each feature extraction process is error prone: quantization partially overcomes this problem if erroneous measurements are reported to the same quantized value of the correct one. Moreover, quantization can be useful when the music signal itself may have variations due to expressive performances, such as note durations played with *rubato* or note pitches played with *vibrato*.

Quantization can also be considered a kind of stemming. In fact, it is well known that many works are based on a limited amount of music material, which is varied and developed during the composition [39]. In this case, the conflation of different thematic variations into a single index will improve recall because the user may choose any of these variations to express the same information need. The increase in recall usually corresponds to a lowering in precision, because a quantized lexical unit is usually more generic and less precisely describes a user information need. To this end, the effect on precision due to a coarse quantization can be reduced using long music excerpts as queries [13].

Quantization can be carried out on any music dimension, and at different levels. Table 1 shows possible approaches to the quantization of melodic intervals that have been proposed in the literature.

Table 1: Number of different symbols when quantization is applied to intervals within an octave (only the names of the ascending ones are given)

Quantization level	Symbols
cents $0, 1, \ldots, 1200$	2401
semitones: $0, 1, \ldots, 12$	25
intervals: *unison*, *second*, …, *octave*	15
perceptual intervals: *unison*, *small*, *medium*, *large*	9
direction: *same*, *up*	3

A similar approach to quantization can be carried out on rhythmic information. It should be noted that score representation itself is a quantized version of possible performances, because playing the exact onset times will result in a "mechanical" performance. In the case of transcriptions of preexisting performances, rhythm quantization is a common practice because the transcriber chooses note onsets as a compromise between the readability of the score and the precision of the reported times. Many approaches to melodic indexing do not take into account note durations, but are based only on pitch information, and they can be considered a limit case where there is only one level of rhythm quantization.

Quantization is a way to deal with musical variants implicitly. In this chapter an approach is proposed which explicitly takes into account differences within melodic phrases by modeling them in a statistical framework. This approach is presented in Sect. 4.

Terms Weighting

As is well known, index terms do not describe the content of a document to the same extent. The importance of a term in describing a document varies along a continuum that ranges from totally irrelevant to completely relevant. For textual information retrieval, it is generally assumed that the frequency at which a word appears in a document is directly proportional to its relevance, while the frequency at which it appears in the collection is inversely proportional to its relevance. These considerations gave birth to a popular weighting scheme, called term frequency - inverse document frequency, in short $tf \cdot idf$, which has been proposed with different variants.

A parallel analysis of units rele. If a musical lexical unit, for any chosen dimension, appears frequently inside a given document, it is very likely that listeners will remember it. Moreover, a frequent lexical unit can be the signature of the style of a composer [8]. Thus, term frequency seems to be a reasonable choice for music documents too. On the other hand, a lexical unit that is very common inside a collection of documents can be related to the style of a thematic collection – the chord progression of blues songs – or can correspond to a simple musical gesture – a major scale – or can be the most used solution for particular passages – the descending bass connecting two chords. Moreover, a user may not use frequent lexical units as parts of his query because it is clear that they will not address any particular document. Thus inverse document frequency seems to be a reasonable choice as well.

However, some care has to be paid to a direct application of a $tf \cdot idf$ weighting scheme to music indexing, at least because users access music documents differently from textual ones. In particular, music documents are accessed many times by users, who may only listen to selected excerpts. Moreover, it is common practice for radio stations to broadcast only the parts of the songs with the sung melody, skipping the intro and the coda. The relative importance by which a lexical unit describes a document should also reflect these aspects, which cannot be inferred by document analysis alone. Moreover, listeners are likely to remember and use in their queries the part of the song where the title or particularly relevant lyrics are sung, which becomes more relevant disregarding its frequency inside the documents and inside the collection. It should be noted that there have been very few studies that investigate the best weighting scheme for music indexing, and in many cases a direct implementation of the $tf \cdot idf$ is used.

The possibility to give different weights to lexical units is a crucial difference between information retrieval and approaches based on approximate string matching techniques. The former allows the documents to be ranked depending on the relevance of their lexical units as content descriptors, while the latter allows the documents to be ranked depending on the degree at which an excerpt of each document matches the query. In other words, a good match with an almost irrelevant excerpt may give a higher rank than a more approximate match with a highly relevant excerpt. A methodology to combine a weighting scheme with approximate matching is presented in Sect. 4.

3 Approaches to Music Retrieval

Searching for a musical work given an approximate description of one or more of its dimensions is the prototype task for a music retrieval system. In principle, retrieval can be carried out on any dimension. For instance, the user could provide an example of the timbre – or of the sound – that he is looking for, or describe the particular structure of a song. However, most of the approaches are based on melody as the main, and often only, content descriptor. This choice depends on the typical interaction paradigm that is used to query a system, called *query by example*, which requires the user to give an example of his information need by singing, humming, or whistling an excerpt of a song.

The research work on melodic retrieval can be grouped depending on the methodologies that have been proposed to compute the similarity between the query and the documents. A classification in three categories is proposed: approaches based on the computation of *index terms*, which play a role similar to words in textual documents, approaches based on *sequence matching* techniques, which consider both the query and the documents as sequences of symbols and model the possible differences between them, and *geometric methods*, which can cope with polyphonic scores and may exploit the properties of continuous distance measures (in particular the triangular inequality) to decrease computational complexity. Of these approaches, the first two are the ones mostly related to the methodology proposed in this paper.

3.1 Melodic Retrieval Based on Index Terms

An example of research work in this group has been presented in [11], where melodies were indexed through the use of N-grams. Experimental results on a collection of folk songs were presented, testing the effects of system parameters such as N-gram length, showing good results in terms of retrieval effectiveness, though the approach did not seem to be robust to decreases in query length. The N-gram approach has been extended in [9] in order to retrieve melodies in a polyphonic score, without prior extraction of the single melodies.

An alternative approach to N-grams has been presented in [32], where indexing was carried out by highlighting musically relevant sequences of notes, called musical phrases. Unlike the previous approaches, the length of indexes was not fixed but depended on the musical context. Phrases could undergo a number of different normalizations, from the complete information on pitch intervals and duration to the simple melodic profile. Segmentation approaches can also be based on recurrent melodic patterns, as proposed in [41] and further developed in [37]. In this latter case, patterns were computed using either only rhythm, or only pitch, or the combined information, and the final retrieval was carried out using a data fusion approach.

An extensive evaluation of segmentation techniques aimed at extracting index terms has been presented in [38]. Experimental results on a collection of about 2300 musical documents in Midi format showed that N-grams are still the best choice for index terms – 0.98 of average precision – and that recurrent patterns were almost comparable to them – 0.96 of average precision. Segmentation approaches based on a priori knowledge of music perception or structure proved to be more sensible to local mismatches between the query and the documents, giving an average precision of about 0.85 in both cases.

3.2 Melodic Retrieval Based on Sequence Matching

The typical application of these approaches is the retrieval of a precise musical work, given an approximate excerpt provided by the user. To this end, a representation of the query is compared with the representations of the documents in the collection each time a new query is submitted to the system. The main positive aspect of these approaches is that they are able to model the possible mismatches between the query and the documents to be retrieved. As is well known from the string processing domain, possible sources of mismatches are insertions and deletions of musical notes. The modification of a note can be considered either as a third source of mismatch or the combination of a deletion and an insertion.

Approximate string matching techniques have been applied to melodic retrieval. One of the first examples was described in [13], where the melodies were represented by three symbols – ascending or descending interval and same note – in order to cope with possible mismatches in pitch between the query and the documents. The work presented in [2] is based on the use of pattern discovery techniques, taken from computational biology, to search for the occurrences of a simplified description of the pitch contour of the query inside the collection of documents. Another approach, reported in [18], applies pattern matching techniques to documents and queries in GUIDO format, exploiting the advantages of this notation in structuring information. Approximate string matching has been used also by [17], adapting the technique to the kind of input provided by the user. The work presented in [20] reports a comparison of different approaches based on a variant of Dynamic Time Warping, with a discussion on computational complexity and scalability of four different techniques. Other examples of sequence matching can be found in [19, 44].

Alternatively to approximate string matching, statistical models have been applied to sequence matching. An application of Markov chains has been proposed in [5] to model a set of themes extracted from musical documents, while an extension to hidden Markov models has been presented in [43] as a tool to model possible errors in sung queries. A mixed methodology has been presented in [21], where the distance function used in a Dynamic Time Warping approach has been computed using a probabilistic model.

Sequence matching techniques are very efficient, with a computational cost for a single comparison that is $\mathbf{O}(m+n)$, where m is the length of the query and n is the size of the document. However, the application of sequence matching may require the sequence representing the query to be compared to all the documents in the collections. Thus the computational cost of a single retrieval is linear with the size of the collection. This clearly implies a low scalability of direct sequence matching. To overcome the problem, pruning techniques have been proposed in the literature. In particular, the approach described in [40] is based on the creation of a tree structure over the collection of documents depending on the melodic similarity between them: comparisons are carried out only along the path, from the root to a leaf, that gives the best sequence matches.

3.3 Melodic Retrieval Based on Geometric Methods

The matching of the query with documents can be computed in a geometric framework. This approach can cope with polyphonic music without requiring prior extraction of the main melody, because the complete score is represented as a set of points, or lines, on a plane: the vertical axis usually corresponds to pitch while the horizontal axis corresponds to time. The same representation applies to queries.

The geometric approach, which has been introduced in [7], is based on the application of a number of translations to the query pattern in order to find the best matches with the geometric representation of each document. Incomplete matches can also be found with a geometric approach, as described in [47] where scores were represented as points on a plane. An extension of the representation of documents is presented in [46], where a polyphonic score is represented as a set of lines on a plane, the position along the time axis and the length of the line are computed from time onset and note duration, respectively. A further improvement has been proposed in [28], where note duration is exploited to create a weight model that penalizes mismatches between long notes.

The computational cost of geometric approaches is $\mathbf{O}(mn \log n)$, where m is the size of the query and n is the size of the score. The increase in computational complexity is compensated by the fact that these approaches can cope with polyphonic scores. As for sequence matching approaches, a retrieval task may require a number of comparisons that is linear with the collection size, if a pruning or indexing technique is not applied.

To this end, an alternative approach for computing the similarity between a bidimensional representation of queries and documents was presented in [45]. The polyphonic scores are represented as weighted points on a plane, where the positions correspond to pitch and onset time of each note, while the weight is computed from note durations. The melodic similarity is computed through two alternative transportation distances, the Earth Mover's Distance and the Proportional Transportation Distance [14]. The Proportional Transportation Distance is a pseudo metric, for which the triangle inequality holds. This property has been exploited to improve retrieval efficiency, because the query is compared only to a reduced set of documents – called *vantage objects* – exploiting the triangular inequality to rule out all the documents that have a distance from the query higher than a given threshold.

4 A Probabilistic Approach to Music Indexing and Retrieval

The methodology presented in this section combines two of the approaches described in Sect. 3, by combining the idea of sequence matching with an indexing scheme. The work is based on two main considerations. On the one hand, indexing techniques are efficient and scalable, but do not take into account the presence of errors in the query, and thus their retrieval effectiveness decreases as the number of errors increases. On the other hand, sequence matching can direct model differences between queries and documents but is characterized by high computational costs, because the complexity is proportional to the collection size. Figure 2 shows an example of common query

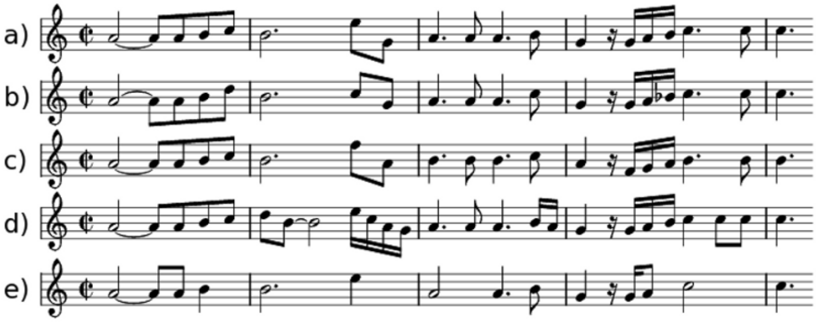

Fig. 2: Common pitch errors in users' queries: **a**) original melody, **b**) pitch errors, **c**) tonality errors, **d**) insertion errors, **e**) deletion errors

errors on pitch, where tonality, insertion and deletion errors are also shown. There may also be errors in note durations, both locally – e.g. the user shortens a long note – and globally – e.g. the user sings faster than the original melody.

With the aim of partially overcoming the drawbacks of both techniques, a novel methodology is proposed for describing music documents with *contour models*, which generalize the concept of lexical units. In particular, contour models are computed from melodic N-grams, that is, on sequences of exactly N notes in the melody, and are used as index terms. Retrieval is then carried out by performing an approximate matching between the lexical units extracted from the query and the contour models computed from the collection. The approximate matching is based on an application of Weighted Transducers (WTs) as models for contours. Once the most probable contour models corresponding to a query are computed, retrieval is then carried out using standard techniques. The methodology can be summarized as follows.

At indexing time:

- all the J sequences of N notes in the collection are extracted and used to build WT models M_j, with $j \in J$;
- M_j are indexed using an inverted file, which links each M_j with the music documents it belongs to.

At retrieval time:

- the user's query is transcribed to a sequence of notes, from which all the subsequences Q_i of N notes are computed;
- the probability $p(i, j)$ that Q_i corresponds to model M_j is computed for all i and j;

- the distance between the query and the documents is computed using the Vector Space Model [1], with a variant of the $tf \cdot idf$ weighting scheme that takes into account the probability $p(i,j)$;
- the distance between the query and the documents is used to build a list of potentially relevant documents in descending order of similarity.

This methodology was implemented in a music retrieval prototype, which was tested on a collection of music documents and with a set of audio queries.

4.1 Description of the Model

The approach is based on the application of a probabilistic model [35] that is defined as follows.

Definition 1. A string-to-weight subsequential transducer, or weighted transducer, is an 8-tuple
$T = (Q, i, F, \Sigma, \phi, \sigma, \lambda, \rho)$, where:

- Q is the set of N states, each one described by a vector,
- i is the initial state,
- $F \subseteq Q$ is the set of final states,
- Σ is the input alphabet,
- ϕ is the transition density functions mapping $Q \times \Sigma$ to $Q^\star$,
- σ is the output function mapping $Q \times \Sigma$ to $\mathbb{R}_+$,
- $\lambda \in \mathbb{R}_+$ is the initial weight,
- ρ is the final weight function mapping F to $\mathbb{R}_+$.

A string w is accepted by a transducer T if there exists $f \in F$ such that $\phi(i, w) = f$. The output associated with w is: $\lambda + \sigma(i, w) + \rho(f)$.

WTs are similar to Hidden Markov Models (HMMs) [3], with the important difference that the transition density function ϕ is time-varying, because it also depends on the input symbol at time t, and has to be recalculated for each symbol $x(t)$ of the input sequence. The output function σ plays a similar role of the emission probability function of HMMs, that is, the probability of observing a given symbol when in a given state, even if the observation probability is defined as a weight.

The three main problems of HMMs [42] – namely recognition of an unknown sequence, decoding of the most probable state sequence given an observation, and parameters training – can be solved for WTs too. In particular, the problem of recognizing an unknown sequence of observations can be solved by calculating the best weight obtained by a legal state sequence of a model given the input sequence X, and can be computed by equation

$$\sigma(i, f) = \max_{\forall s_k \in Q \mid s_1 = i, s_N = f} \sigma(S|X) \tag{1}$$

where S is the sequence of states $s_1, s_2, \ldots, s_N$ and s_k is its generic element.

The recognition of an unknown input sequence is carried out by computing the probability of the most probable path corresponding to the sequence of observations, using the iterative steps:

$$\pi_k(j) = \max_{[s_1 s_2 \ldots s_{k-1}] \subset Q^*} P[s_1 s_2 \ldots s_k = j | T,\, x_1 x_2 \ldots x_k],\ j \in Q \tag{2}$$

$$\pi_{k+1}(j) = \left[\max_i \pi_k(i)\phi_{ij}(x_k)\right] \cdot \sigma_j(x_{k+1}),\ 1 \leq k \leq input\, size \tag{3}$$

The iteration of these two steps over the input sequence X_1^K gives the best state path and the weight of the model conditioned by the input. The weight corresponds to the weight of the last state in the sequence.

4.2 Application of Weighted Transducers to Melody Representation

Once the model and the methods to perform a recognition task have been defined, they can be applied to model melodic contours for a music retrieval task, as described in this section.

The information used to describe a melody is bases on pitch and the duration of events (notes and rests) in the score and in the query. A preprocessing step transforms this initial information in the array [*duration, Δpitch*], where duration is represented in beats and Δpitch(k) = $pitch(k) - pitch(k-1)$, where $pitch(k)$ is the MIDI pitch of note k and hence Δpitch is the difference in semitones between two subsequent notes. Rests are represented with a conventional $\Delta pitch = -50$, which is a value very unlikely to be found in any score.

The input alphabet of the WT is composed by a vector Σ in which the measured duration and the measured Δpitch of the current note in the query are nested.

$$\Sigma = \begin{bmatrix} measured\ duration \\ measured\ \Delta pitch \end{bmatrix} \tag{4}$$

The variables describing the states are based on the computation, from the score, of the *nominal duration* and *nominal Δpitch*, to which the *variance duration*, conventionally set to $\frac{1}{5}$ of the nominal duration, and *variance Δpitch*, conventionally set to 0.6 semitone, are added. The *variance* are the parameters of independent Gaussian probability distributions, which are centered in *duration* and *Δpitch*, respectively. Finally, a fifth parameter called *distance factor* is added, which is a number that gives a low probability to transitions towards events distant in the score. The distance factor is heuristically set by simply halving the probability each time an event is skipped by a transition.

A generic state Q is then described by the vector.

$$Q = \begin{pmatrix} nominal\ duration \\ nominal\ \Delta pitch \\ variance\ duration \\ variance\ \Delta pitch \\ distance\ factor \end{pmatrix} \tag{5}$$

where every model state Q corresponds to one contour symbol, that is, to an event in the melody.

Regarding the state topology of the WT, in the proposed application the contour is modeled with only one initial and one final state, giving a trivial definition of functions ρ and λ. The model is automatically built by assigning a different state for each note in the contour, considering that the last note state is also the final state. A non-zero transition probability is assigned between states associated with subsequent notes, while other transitions are placed to admit hops of one or two notes in the original score. Figure 3 depicts a model for a sequence of four notes, where S_0 is the initial state, and states $S_1 \dots S_4$ correspond to the notes in the contour, with S_4 also the only final state.

From each melody in the collection, the system creates a set of reference models. Each model is representative of a contour subsequence of a fixed number of events. Events duration is always normalized to the length of the complete subsequence, to deal with different timings between the documents and the query. Local mismatches are taken into account by the modeling of durations with a Gaussian distribution centered on the nominal duration. The output function σ is given directly by Viterbi decoding, by computing the weights on the best state path. As a consequence, the transition state function ϕ subsumes all the relevant information on the model. As for HMM, ϕ can be given by its transition state matrix $\phi = \{\phi_{ij}(x)\},\ i, j \in Q$.

The choice of the proposed modeling is motivated by the following characteristics:

- WTs are *stochastic* – it is not known a priori whether the input symbol is a correct note or an error and alternative transitions are taken into account;
- the models are *Markovian* with a *left-to-right* topology – psychological studies [10] showed that melodies are remembered by pitch intervals between successive notes;
- WTs are *time-varying* – the local similarity between the input sequence and the contour may change, depending on previous input, because errors in pitch or duration often affect pitch and duration of subsequent notes.

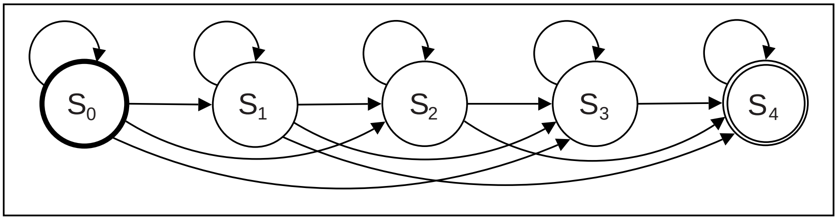

Fig. 3: Weighted Transducer that represents a four notes contour. Arrows are drawn when the transition probability is non zero

It should be noted that this modeling is adequate to manage all query errors shown in Figure 2. For instance, deletion errors of one or two notes are managed by transitions that jump some states in the longest path from i to f, and local errors in pitch and duration are directly modeled by the stochastic nature of the WT. Moreover, insertion errors find a correspondence in autotransitions. Tonality errors and time stretching errors are compensated by adoption of $\Delta pitch$s and by time relativization on every single contour.

4.3 Computation of the Similarity between a Query and the Documents

Each document in the collection is described by its set of WT models, that are computed from its contours. This representation is the basis for the computation of the similarity between users' query and the documents in the collection.

For each query subsequence s, of the same length of the contours, the weight $v(s, m)$ corresponding to each model m is computed using the Viterbi algorithm described in Sect. 4.1. The output of this step was used to compute a measure of similarity between the query and the documents, using a variant of the $tf \cdot idf$ weighting scheme.

The standard $tf \cdot idf$ computation is based on equation 6 where $freq(m, d)$ is the frequency of model m in document d, N is the number of documents in the collection, n_m is the number of documents that contain contour m.

$$t_{m,d} = \frac{freq(m,d)}{\max_k freq(m,k)} \times \log \frac{N}{n_m} \tag{6}$$

The $tf \cdot idf$ measure is based on an exact match between query and document terms, this way the value of $t_{m,d}$ can be computed in the same way both for documents and query terms. Given the approximate match due to WTs modeling, the information $v(s, m)$ about the similarity between query sequence and document models has to be added to the model. This is achieved by the following equation:

$$sim(q,d) = \frac{\sum_{s \in q, m \in d} v(s,m) \times occ_q(s) \times t_{m,d}}{\sqrt{\sum_{s \in q} (v(s,m) \times occ_q(s))^2}} \tag{7}$$

where $occ_q(s)$ is the frequency of sequence s in query q. The denominator is used as a scaling factor to have similarities in the same range. This approach, in its classical version, considers all the models m that match a document term. However, it can be modified to speed up computation time, by considering only the first $Nmax$ models m, ordered by the value of $v(s, m)$.

5 Experiments

The approach was tested using a small collection of 2004 pop songs in MIDI format. The monophonic melody was extracted from each song and used to build the set of WT models. The choice of using monophonic representation depends on the fact that queries are usually monophonic melodies sung by users. In the experiments, contours with a fixed length, from three to seven notes, were used. According to the discussion in Sect. 2, this choice corresponds to using an N-gram approach on the melodic dimension, with different values of N, and considering as stopwords all the sequences with a different length. The choice of this range of N for testing the approach was motivated by the fact that shorter sequences – two or one notes – were considered too generic to be a good content descriptor, while longer sequences – more than seven notes – increased the computational complexity.

Small models are likely to overlap among documents and then the number of new models for each new document is likely to increase sublinearly. In order to reduce the number of models, two contours with the same pitch interval sequence and with durations that differ by of 3% at most were associated with the same model. Figure 4 shows the characteristics of the WT models set depending on the number of note events, one for each row. The first column represents the total number of models for each song, the second the number of new models for each song normalized by the number of models that represent that song, while the last column represents the increase in the number of different models versus the increase in the number of songs.

The most noteowrthing result is given by the last column graphs which shows that for three note models the WT models set size increases sublinearly because of the overlapping of models between documents. The model overlap decreases with the increasing of the model size, down to 1.15 for seven note models. Since this ratio depends on the song number, it is reasonable to assume that the overlap will increase with more complete song collections.

The relationship between the length of the contour and the number of overlapping models is summarized in Table 2.

The queries used in the experiments have been provided by the MAMI team [29], which worked on an experiment on query by singing [27]. In particular, a set of 31 queries were used, for which MAMI experts produced a manual pitch-tracking. The choice of manual transcriptions is due to the fact that the main interest of this work is to model the differences between the melodies in the documents and the melodies provided by the users, while the possible mismatches due to automatic transcriptions are not taken into account. The input of the WTs was extracted from contours of transcribed queries with the same approach described for document contours. Query contours were joined together when the length and the $\Delta pitch$ of each note did not differ over 10% and 0.3 semitones, respectively.

Five tests were carried out, one for each model size from 3 to 7 contour symbols. To this end, the collection was indexed by five WT model sets using

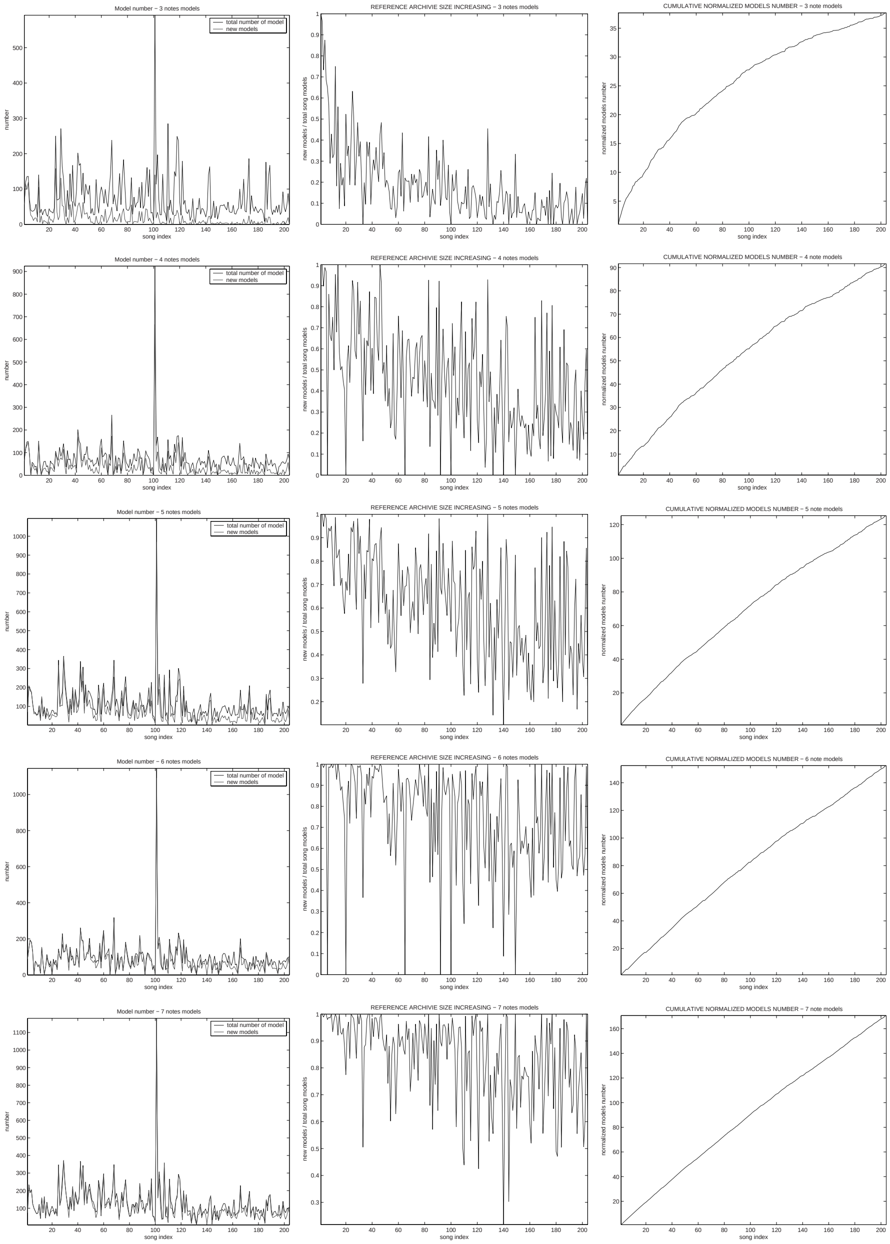

Fig. 4: Number of different models per document (left); number of new models for each new document (middle); number of different models versus number of documents (right) – rows correspond to the increase in the contour length, from 3 to 7 notes

Table 2: Characteristics of the index size depending on the length of the contours

N	models	overlap
3	3485	4.40%
4	7591	1.91%
5	15887	1.45%
6	15911	1.26%
7	23361	1.15%

all the 2004 different music documents, with one index for each model size. The retrieval procedure described in Sect. 4.3 was applied, with different values for $Nmax$. The best retrieval performances were obtained using $Nmax$ in the interval [3, 5]. In general a small $Nmax$ value gives a more selected result but reduce the recall.

Results are reported in Table 3, which shows encouraging results obtained with $Nmax = 3$. As expected, models with longer contours improve the average precision because they exploit more information about the melody. The three note model set has a low performance with only 14% average precision which denotes the inadequacy of short models. Increasing the model size, with only four symbol contours the retrieval system produces a substantial performance increase with seven right songs in the first position. The quality of results increase to 50% average presision with seven note models. An unexpected outcome is given by the six note model set, which produces a retrieval precision lower than the five note model set. It is clear that the recall level does not present a variation trend increasing the models size. The number of songs found in the first 50 positions is nearly constant.

These results can be compared with other approaches applied to test collections of similar size. For instance, a HMM-based indexer [43] retrieved the correct song with the highest rank 41.7% of times. In the same paper, a simple string matcher is used as a baseline, giving a value of 16.7% for the same experiment. The approaches based on Dynamic Time Warping [20] gave, in the best case, the relevant document retrieved at top rank 54.0% of times. The results with Markov models [5] showed that in 60.0% of cases the correct song

Table 3: Retrieval performances

N	$= 1$	≤ 3	≤ 10	≤ 50	precision
3	6.45%	12.90%	35.48%	58.06%	14.3%
4	22.58%	38.71%	51.61%	83.87%	35.0%
5	32.26%	38.71%	64.52%	83.87%	40.4%
6	29.03%	38.71%	54.84%	77.42%	37.4%
7	45.16%	48.39%	58.06%	80.65%	50.0%

was the nearest neighbor of the query. It should be noted that the approaches presented in the literature have a complexity that is linear with the number of documents, while the approach described in this chapter has a complexity that is linear with the number of contour models. All the approaches were tested with real audio queries, while higher precision and recall are expected when symbolic queries are used.

It can be interesting to compare these results with the ones obtained with the same query set, but with a larger test collection [38] using different segmentation techniques combined with a text retrieval approach. When N-grams were used to segment the melodies, the percentage of times the correct song was retrieved at top rank was only 12.6%. A small increase, up to 15.1%, could be obtained using melodic patterns instead of N-grams. This result may show that combining the segmentation with a probabilistic match may considerably improve retrieval effectiveness.

6 Conclusions

This chapter presents an overview of problems and characteristics of music retrieval, and presents a novel methodology based on approximate indexing of music documents. The basic idea is to merge the positive effects of document indexing in terms of efficiency and scalability, with the positive effects of approximate matching in terms of robustness to local mismatches. The methodology was tested on a test collection of music documents, using a set of transcribed audio queries, with encouraging results.

References

1. Baeza-Yates, R., Ribeiro-Neto, B. (eds.): Modern Information Retrieval. ACM Press, New York, NY (1999)
2. Bainbridge, D., Nevill-Manning, C., Witten, I., Smith, L., McNab, R.: Towards a digital library of popular music. In: Proceedings of the ACM Conference on Digital Libraries, pp. 161–169 (1999)
3. Bengio, Y.: Markovian models for sequential data. Neural Computer Surveys **2**, 129–162 (1999)
4. Berenzweig, A., Logan, B., Ellis, D., Whitman, B.: A large-scale evaluation of acoustic and subjective music-similarity measures. Computer Music Journal **28**(2), 63–76 (2004)
5. Birmingham, W., Dannenberg, R., Wakefield, G., Bartsch, M., Bykowski, D., Mazzoni, D., Meek, C., Mellody, M., Rand, W.: MUSART: Music retrieval via aural queries. In: Proceedings of the International Conference on Music Information Retrieval, pp. 73–82 (2001)
6. Cantate: Computer Access to Notation and Text in Music Libraries (July 2007). `http://projects.fnb.nl/cantate/`

7. Clausen, M., Engelbrecht, R., Meyer, D., Schmitz, J.: PROMS: A web-based tool for searching in polyphonic music. In: Proceedings of the International Symposium of Music Information Retrieval (2000)
8. Cope, D.: Pattern matching as an engine for the computer simulation of musical style. In: Proceedings of the International Computer Music Conference, pp. 288–291 (1990)
9. Doraisamy, S., Rüger, S.: A polyphonic music retrieval system using N-grams. In: Proceedings of the International Conference on Music Information Retrieval, pp. 204–209 (2004)
10. Dowling, W.: Scale and contour: Two components of a theory of memory for melodies. Psychological Review **85**(4), 341–354 (1978)
11. Downie, S., Nelson, M.: Evaluation of a simple and effective music information retrieval method. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp. 73–80 (2000)
12. Dunn, J., Mayer, C.: VARIATIONS: A Digital Music Library System at Indiana University. In: Proceedings of ACM Conference on Digital Libraries, pp. 12–19 (1999)
13. Ghias, A., Logan, J., Chamberlin, D., Smith, B.: Query by humming: Musical information retrieval in an audio database. In: Proceedings of the ACM Conference on Digital Libraries, pp. 231–236 (1995)
14. Giannopoulos, P., Veltkamp, R.: A pseudo-metric for weighted point sets. In: Proceedings of the European Conference on Computer Vision, pp. 715–730 (2002)
15. Gómez, E., Herrera, P.: Estimating the tonality of polyphonic audio files: Cognitive versus machine learning modelling strategies. In: Proceedings of the International Conference on Music Information Retrieval, pp. 92–95 (2004)
16. Harte, C., Sandler, M., Abdallah, S., Gómez, E.: Symbolic representation of musical chords: a proposed syntax for text annotations. In: Proceedings of the International Conference on Music Information Retrieval, pp. 66–71 (2005)
17. Haus, G., Pollastri, E.: A multimodal framework for music inputs. In: Proceedings of the ACM Multimedia Conference, pp. 282–284 (2000)
18. Hoos, H., Renz, K., Görg, M.: GUIDO/MIR – an experimental musical information retrieval system based on GUIDO music notation. In: Proceedings of the International Symposium on Music Information Retrieval, pp. 41–50 (2001)
19. Hsu, J.L., Liu, C., Chen, A.: Efficient repeating pattern finding in music databases. In: Proceeding of the International Conference on Information and Knowledge Management, pp. 281–288 (1998)
20. Hu, N., Dannenberg, R.: A comparison of melodic database retrieval techniques using sung queries. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, pp. 301–307 (2002)
21. Hu, N., Dannenberg, R., Lewis, A.: A probabilistic model of melodic similarity. In: Proceedings of the International Computer Music Conference, pp. 509–515 (2002)
22. Huron, D.: The Humdrum Toolkit: Reference Manual. Center for Computer Assisted Research in the Humanities, Menlo Park, CA (1995)
23. Jones, K.S., Willett, P.: Readings in Information Retrieval. Morgan Kaufmann, San Francisco, CA (1997)
24. Krumhansl, C.: Why is musical timbre so hard to understand? In: S. Nielsen, O. Olsson (eds.) Structure and Perception Electroacoustic Sound and Music, pp. 45–53. Elsevier, Amsterdam, NL (1989)

25. Lee, J., Downie, J.: Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In: Proceedings of the International Conference on Music Information Retrieval, pp. 441–446 (2004)
26. Lesaffre, M., Leman, M., Tanghe, K., Baets, B.D., Meyer, H.D., Martens, J.P.: User-dependent taxonomy of musical features as a conceptual framework for musical audio-mining technology. In: Proceedings of the Stockholm Music Acoustics Conference, pp. 635–638 (2003)
27. Lesaffre, M., Tanghe, K., Martens, G., Moelants, D., Leman, M., Baets, B.D., Meyer, H.D., Martens, J.P.: The MAMI query-by-voice experiment: Collecting and annotating vocal queries for music information retrieval. In: Proceedings of the International Conference on Music Information Retrieval, pp. 65–71 (2003)
28. Lubiw, A., Tanur, L.: Pattern matching in polyphonic music as a weighted geometric translation problem. In: Proceedings of the International Conference of Music Information Retrieval, pp. 289–296 (2004)
29. MAMI: Musical Audio Mining – "query by humming" (July 2007). `http://www.ipem.ugent.be/MAMI/`
30. McLane, A.: Music as information. In: M. Williams (ed.) Arist, Vol. 31, chap. 6, pp. 225–262. American Society for Information Science (1996)
31. Meek, C., Birmingham, W.: Automatic thematic extractor. Journal of Intelligent Information Systems **21**(1), 9–33 (2003)
32. Melucci, M., Orio, N.: Musical information retrieval using melodic surface. In: Proceedings of the ACM Conference on Digital Libraries, pp. 152–160 (1999)
33. Melucci, M., Orio, N.: A comparison of manual and automatic melody segmentation. In: Proceedings of International Conference on Music Information Retrieval, pp. 7–14 (2002)
34. Middleton, R.: Studying Popular Music. Open University Press, Philadelphia, PA (2002)
35. Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics **23**(2), 269–311 (1997)
36. Musica: The International Database of Choral Repertoire (July 2007). `http://www.musicanet.org/`
37. Neve, G., Orio, N.: Indexing and retrieval of music documents through pattern analysis and data fusion techniques. In: Proceedings of the International Conference on Music Information Retrieval, pp. 216–223 (2004)
38. Orio, N., Neve, G.: Experiments on segmentation techniques for music documents indexing. In: Proceedings of the International Conference on Music Information Retrieval, pp. 104–107 (2005)
39. Owen, G.: Using connectionist models to explore complex musical patterns. Computer Music Journal **13**(3), 67–75 (1989)
40. Parker, C.: A tree-based method for fast melodic retrieval. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, pp. 254–255 (2004)
41. Pienimäki, A.: Indexing music database using automatic extraction of frequent phrases. In: Proceedings of the International Conference on Music Information Retrieval, pp. 25–30 (2002)
42. Rabiner, L.: A tutorial on hidden Markov models and selected applications. Proceedings of the IEEE **77**(2), 257–286 (1989)
43. Shifrin, J., Pardo, B., Meek, C., Birmingham, W.: HMM-based musical query retrieval. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, pp. 295–300 (2002)

44. Tseng, Y.: Content-based retrieval for music collections. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp. 176–182 (1999)
45. Typke, R., Veltkamp, R., Wiering, F.: Searching notated polyphonic music using transportation distances. In: Proceedings of the ACM International Conference on Multimedia, pp. 128–135 (2004)
46. Ukkonen, E., Lemström, K., Mäkinen, V.: Geometric algorithms for transposition invariant content-based music retrieval. In: Proceedings of the International Conference of Music Information Retrieval, pp. 193–199 (2003)
47. Wiggins, G., Lemström, K., Meredith, D.: SIA(M)ESE: An algorithm for transposition invariant, polyphonic content-based music retrieval. In: Proceedings of the International Conference of Music Information Retrieval, pp. 283–284 (2002)