C-151

Manipulation of Music For Melody Matching

Alexandra. L. Uitdenbogerd *Justin Zobel*

Computer Science Department
Royal Melbourne Institute of Technology
Melbourne, Victoria, Australia

sandrau@rmit.edu.au *jz@cs.rmit.edu.au*

Keywords Database systems, music databases, approximate matching.

Abstract

Large volumes of music are available online, represented in performance formats such as MIDI and, increasingly, in abstract notation such as SMDL. Many types of user would find it valuable to search collections of music via queries representing music fragments, but such searching requires a reliable technique for identifying whether a provided fragment occurs within a piece of music. The problem of matching fragments to music is made difficult by the psychology of music perception, because literal matching may have little relation to perceived melodic similarity, and by the interactions between the multiple parts of typical pieces of music. In this paper we analyse the properties of music, music perception, and music database users, and use the analysis to propose alternative techniques for extracting monophonic melodies from polyphonic music; we believe that such melodies can subsequently be used for matching of queries to data. We report on experiments with music listeners, which rank our proposed techniques for extracting melodies.

1 Introduction

Content-based retrieval is being explored for many different forms of media. In the case of audio data, the content may consist of sounds, speech, or music, the topic of this paper. Music can be stored in several ways, depending on the use to which the data will be put: represented as a wave-form, as sheet-music, or as performance information.

Content-based retrieval of music has many applications, from casual users wanting to identify a nagging fragment of music to artists verifying that a new composition is indeed original. A likely form of query for such retrieval is a short melody fragment, input via a keyboard or possibly even hummed or sung [11, 17]. The task of a retrieval system is to identify the pieces of music that contain music fragments that sound similar to the query. To achieve this task requires a technique for matching melody fragments. By analogy with the conceptually similar problem of text retrieval [19], the matching technique need not be perfect—users are likely to be tolerant of sloppiness in matching—but it needs to be reasonably reliable, so that users can find matches fairly rapidly. We examine possible ways of matching pieces of music, including the sequences of notes, intervals between the notes, and melody contour. We argue that a combination of techniques is likely to be the most effective solution: evidence of similarity does not solely depend on the notes played. However, a successful indexing technique needs to account for both user needs and the psychology of music perception. For example, when two notes are played simultaneously (either

in a chord or in separate parts) it is usually the case that the upper note is perceived as carrying the melody.

In this paper we examine properties of music, properties of music perception, and likely uses of music databases, and use these factors to propose several alternative methods for extracting monophonic melodies from polyphonic music, thus allowing searching: previous work on music searching has assumed that the music is monophonic, and indeed it is far from clear how searching of polyphonic music might proceed. We test our melody extraction techniques with experiments with listeners, asking them to identify which of the extracted melodies most resembles the original piece. These experiments, although limited in scope, identify a clear winner, which surprisingly is the most naive of the methods we considered.

We concentrate on music performance information, that is, a set of notes that have been played in a given time sequence. However, the same techniques could readily be applied to abstract music notation.

2 Basic Music Concepts

Each time a key is pressed on a piano, a bow is struck across a violin string or air is blown across a flute mouthpiece to make a sound, we say that a note has been played. Musicians do not usually discuss the frequencies of notes, but refer to them as notes of a particular pitch. For example, the note A to which an orchestra is tuned is defined to have a frequency of 440 Hz. The pitch distance between notes is an interval. If one note has double the frequency of another, then it is said to be one octave above the lower note, that is, the interval between the notes is one octave. Notes that are a whole number of octaves apart have the same note name, which is usually a letter, or a letter combined with a sharp (♯) or flat (♭) symbol. For example, the note A with frequency 440 Hz is one octave lower than the note A with frequency 880 Hz.

The notes used in the music of European cultures result from dividing up an octave into 12 notes that are a semitone apart. Although the octave is divided into 12 notes most simple tunes use only a subset of them. For example, the C major scale consists of the notes

C D E F G A B C.

Melodies that use these notes exclusively are said to be in the key of C major. The G major scale consists of the notes

G A B C D E F♯ G.

Music that largely consists of notes from a particular key is *diatonic*. Music that largely ignores keys is *atonal*. Melodies can be *transposed* from one key to another, preserving the intervals between notes. The melody is preserved, and may sound a little higher or lower than the original, but many listeners will not distinguish between the original and the transposition.

When more than one note is played simultaneously, the result is a *chord*. Music that is played along with the main melody is called the accompaniment, which may consist of chords and possibly percussion, or may be a more complex mixture that includes counter-melodies. There may be more than one musical instrument involved, in which case, the term *part* refers to the music for a particular instrument. Music that consists of a melody with no accompaniment is *monophonic*. In this paper we use monophonic in a stricter sense, to refer to music in which no more than one note is sounded at any given time; other music is *polyphonic*.

Music is written on a staff consisting of five horizontal lines, where horizontal position indicates order of play and vertical position indicates pitch. For example, the scale of C major and the C major chord are shown in Figure 1. Music is normally divided into bars, each containing the same number of beats. The number of beats per bar and the quality of the beats is indicated by the time signature.

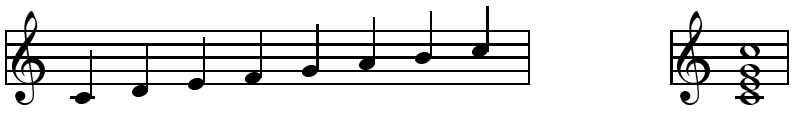

Figure 1: *Music notation. On the left, the C major scale. on the right, a C major chord.*

3 Music Databases

Being able to retrieve data from a music database using a melody fragment as a query would be desirable for several applications. It is useful for copyright searches, locating a particular musical work for which one doesn't have the title, performing musicological studies of music collections, and as a composition tool. The needs of users of music databases differ depending on the purpose of their queries. In the case of copyright searches, one would need to be able to retrieve all works that are sufficiently similar to the query. In the case of the user who is locating a particular work which they can only remember a part, only one answer is being searched for.

Types of user

Users such as purchasers of music or library patrons would usually be interested in a specific piece of music of which they can recall only a fraction, which might be a melody, a fragment of the accompaniment, or even a part that transfers from one instrument to another. To these users, a query result will only be relevant if it is the same as the musical work that they remember. The other results may have some similarity that leads them to be considered relevant in a general sense, but will not be of use to the user; this difference between "topicality" and "utility" is discussed by Blair [4].

Composers or songwriters may be concerned about whether their work infringes the copyright of existing works—the nature of the process of composition makes it difficult to distinguish between an inspired original musical idea and a musical theme that was once heard and half remembered. A composer's query may consist of a fragment of melody that they wish to verify is not in infringement; or it may also consist of a completely arranged piece of music; or may fall somewhere in between these two extremes. Composers may also be interested in types of queries that aid the composition process. If a melody is being harmonised or arranged, a composer might like to see how other composers handled a particularly difficult sequence of notes. For long musical works, the location within the work would allow the user to locate the relevant part of the work.

Different types of users have different levels of skills when it comes to describing their query. Composers, musicologists and other musicians could be expected to be able to prepare a query using a keyboard or notation with a good degree of accuracy. The lay person would have some difficulty in preparing a music query. Several systems have been described that allow the user to perform queries by humming or singing [11, 14, 17]. Problems with this method is that people do not sing accurately, especially if they are inexperienced or unaccompanied; even skilled musicians have difficulty in maintaining the correct pitch for the duration of a song. However, melody contour, that is, the pitch direction of the notes, is usually accurate.

Musical Data

Music can be represented in many ways. It can be stored as: a waveform representing the sound, an image of the sheet music, performance information, or music notation layout information. Some formats such as the draft standard SMDL (Structured Music Description Language) [13] allow for all of these. The MIDI (Musical Instrument Digital Interface) standard is a commonly-used format for musicians who exchange music sequences and for lay people who play music on their computers. In contrast to SMDL, it usually includes performance information only, that is, when

a note is played, how loudly, and for how long. It can only be used for instrumental music and does not store notation information such as the division of music into bars or measures. Standard MIDI files contain one or more tracks of MIDI events. Most MIDI files contain music that is to be played on more than one instrument. Usually each instrument has a separate track.

The Internet has made large quantities of MIDI data available. One site archives MIDI files sent to a newsgroup and contains about 15 000 files. MIDI data from the Internet is used as a basis for the music data analysis in this paper.

4 Music Perception

Several aspects of music perception need to be considered when developing matching techniques for music databases. First is the type of query that users will present. In the case of someone trying to locate a half-remembered fragment of music, it is useful to understand how people remember music and in particular, how they remember melodies. Second, since most music that we hear contains both melody and accompaniment, it is necessary to determine what would be perceived as melody in an accompanied musical work. Third, since many music queries involve finding similar but not exact matches to melodies, we need to decide what similarity means in terms of music perception. We now discuss these areas and the implications for music databases.

Music Memory

There has been much research on how people build mental structures while listening to music and on how music is remembered. Dowling [8] discovered that melody contour is easier to remember than exact melodies. Contour refers to the shape of the melody, indicating whether the next note goes up, down, or stays at the same pitch. He postulates that contour and scale are stored separately in memory. Subjects found it easier to distinguish between a diatonic melody and an atonal melody with the same contour, than between a diatonic melody and a copy that stays within the same scale. The easiest task of all was to distinguish between a melody and another that did not preserve contour. This experiment only tested short-term memory.

Dowling [8] discusses several other experiments that have been performed to determine how musical memory operates. These include long-term memory experiments by Attneave and Olsen, which showed that people do distinguish between a well-known melody and inexact tonal copies of it. Results of other experiments by Dowling and Fujitami show that a copy of a known melody that preserves the contour, but not the interval size, is slightly easier to recognise if the relative interval sizes remain the same. Deutsch showed that changing the octave of notes in a melody makes it hard to recognise. Dowling and Hollombe, and Idson and Massaro showed that it is slightly easier if the contour remains the same. This is a surprising result, given that notes of the same octave are usually considered to be harmonically identical.

As with most music perception studies there are significant differences between the results for highly experienced musicians, people with some music experience, and those with no or little music training. Memory tasks are generally performed better by experienced musicians than by those with less experience.

Figure and Ground

In order to extract melodies reliably from polyphonic music files it is necessary to determine what a person listening to the music would perceive as the melody. Several papers have explored the way that groups of notes are perceived.

Francès [10] investigates the Figure and Ground relationship for music. Several factors are discussed. A melody is heard as the *figure* if it is higher than the accompanying parts. However, if the upper notes are constant and the lower notes form a more interesting pattern, then the lower notes will be heard as the figure. Other factors that can affect a part being perceived as the figure are its intensity and continuity. The way the music is perceived is not always constant, however.

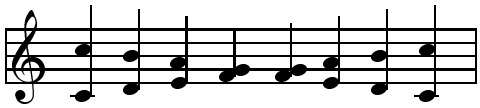

Figure 2: *The scale illusion. Two overlapping scales, one going up and the other going down. The listener hears the upper notes as one part and the lower notes as another part.*

Figure 3: *Rapid sequences of notes in more than one frequency range are perceived as separate parts.*

A listener can make attentional shifts between different parts of the music. Francès observed that experienced musicians can make more rapid attentional shifts than non-musicians, allowing them to be more aware of the accompaniment in a piece of music.

Deutsch [7] discusses the main principles of perception of groups and showed the results for the perception of music. There are four main principles of group perception: proximity, similarity, good continuation, and common fate. The proximity principle states that we group items, that is, we see them as a unit, if they are close together. In the same way, we group items that are similar in some way, or continue in the same direction or end together. These have been clearly shown for visual perception and apply to perception of groups of notes as well. There is a clear hierarchy amongst these principles. Proximity of notes is more important than good continuation as illustrated by the "scale illusion" experiment: if one part is descending in a scale and another part ascends so that they overlap, as shown in Figure 2, listeners perceive the upper notes as one part and the lower notes as a second part (if it is perceived at all). The amplitude or loudness of notes was found to be fairly unimportant in the perception of musical parts compared to proximity, but is also used for grouping of notes. The similarity principle for groups of notes can involve the timbre of the notes. Those that have a similar timbre will be grouped. The timbre is less important than the proximity of notes, however, as shown by Butler and discussed by Deutsch.

If a melody consists of rapid notes where the alternating notes are of a different frequency range, as illustrated in Figure 3, then two musical parts are perceived. This does not occur to the same extent when the melody is slowed down. There is a considerable overlap in terms of how the music will be perceived, so that a melody could be perceived as being two musical parts or just one over a range of speeds.

The research summarised above suggests that melodies extracted from a musical work should take note of the speed of notes as well as the relative frequencies. It may need to extract multiple melodies from a single work in cases where different people may perceive the melody differently.

Melody Similarity Perception

Some aspects of melody similarity were discussed above in the section on memory. Two melodies that have the same contour are perceived as more similar than those that have a different contour. Those that have the same tonality and contour are perceived as more similar than those with the same contour and different tonality (atonal). Exact transpositions of a melody are perceived as the most similar. Van Egmond and Povel [21] discovered that an exact transposition with one note

altered will be considered more similar to the original melody if the note is altered "chromatically" than if it is altered diatonically. A chromatically altered note is one that has been changed by a semitone in such a way that the letter-name remains the same. For example, the note F can be chromatically changed to F♯. Key distance is another factor affecting similarity perception. A key is closely related to another if it differs in only one note. Distant keys are those with few, if any, notes in common. Melodies with the same contour and the same key are perceived as more similar than those in distant keys [3].

Music experience of the subject affects how some melodies will be perceived. Krumhansl and Shepard [15] found that musical listeners are more likely to consider notes that are harmonically similar, whereas non-musical listeners will consider notes of a similar pitch to be similar.

It is highly likely that many melodies will match a user's query, so a similarity ranking based on the above perception results may be useful. Our analysis of the results discussed above suggest the following ordering of factors in music similarity, from good evidence of similarity to poor.

1. Exact transposition in the same key.
2. Exact transposition in a closely related key.
3. Exact transposition in a distant key.
4. Exact transposition except for a chromatically altered note.
5. Exact transposition except for a diatonically altered note.
6. Same contour and tonality.
7. Same contour but atonal.
8. Same note values but different contour.
9. Different contour.

The first three items can be condensed into one, since relative intervals are appropriate for melody comparisons. The music perception survey performed thus far has not revealed any comparisons of rhythm or timing in the perception of similarity. Our experience suggests that difference on a stressed note is more significant than difference on an unstressed note.

Music Perception and Musicology

Theories about music and harmony have a long history, but research on music and its perception began only this century. Music perception research has validated some music concepts, for example the perceived similarity of exact transpositions, but in other cases they are in disagreement. For example, as pointed out by Butler (and discussed by Deutsch [7]), Tchaikovsky's sixth symphony has the theme and accompaniment distributed between two violin parts, but the theme is perceived to come from one set of instruments and the accompaniment from another. This has implications for algorithms that try to extract a melody or convert a collection of notes into musical parts based on music perception. For example, the result of the melody extraction process may not be as originally organised by a composer, even if it correctly simulates how it will be perceived. Conversely, if the stored music is based on the sheet music for a composition, then the musical work may not be retrieved given the user's perception of the melody.

5 Extracting Melodies from MIDI Files

To use a melody to search music, it is necessary to identify the notes in the music that constitute the melody. As discussed earlier, MIDI files can consist of tracks containing notes from many musical instruments playing simultaneously, just as in music notation each instrument has its own

staff. The individual instruments can be playing chordal parts, consisting of several simultaneous notes; there may be more than one "counter-melody" occurring simultaneously; and the perceived melody can move from instrument to instrument. Additionally, in MIDI files some "notes" are actually percussion and therefore have no pitch. That is, the melody may not be present in any individual part—it is far from clear which sequence (or sequences) of notes should be used in the comparison. Extraction of all sequences would lead both to combinatorial explosion and large numbers of sequences with little perceived connection to the original music; indeed, most such sequences would sound like no more than a random series of notes.

The purpose of a melody extraction technique is to identify sequences of notes that are likely to correspond to the perceived melody; given the volumes of music available online, it is not practicable to do this by hand. In terms of the needs of the user, a user may wish to query on a distinctive musical pattern that occurs in an accompanying part. For this reason it may also be useful to retain all parts.

There does not appear to have been much research on the problem of extracting a melody from a piece of music. The approach of Ghias et al. [11] was to ignore percussion and apply other simple heuristics (not described in the paper) to collect melodies. Several papers have discussed splitting polyphonic music into its parts, usually for a set of music with a fairly uniform style [5]. Marsden et al. apply the proximity principle in a rule-based approach, and attempt to refine the rules to split the parts of a Bach fugue [16]. Charnasse and Stepien [5] allocated notes to parts when transcribing German lute tablature. Their approach was also rule-based, mainly using proximity. The number of parts was estimated from chords in the music and the actual parts were produced by processing musical chunks delimited by chords that contain the maximum number of voices.

To obtain melodies, we trialled several approaches that make use of music perception principles: the highest musical part is usually perceived to be the melody, and notes that are close together in pitch are usually considered to belong to the same musical part. We also used first-order predictive entropy as a measure of how interesting a part was, to see if this could be used to predict melody. This calculation is in effect based on the probabilities in a simple Markov model with one state for each note. In highly repetitive tracks such as some forms of accompaniment, each note is reliably predicted by its predecessor, giving a low entropy by this measure, whereas in more varied tracks the entropy is high.

Four methods of extracting a monophonic melody from a MIDI file were developed. All the methods made a single pass through the note stream to select the melody notes, ignoring the note events from channel 10 as these are percussion events in standard MIDI files. In some circumstances, the melodies generated are identical. This occurs when the music consists of a melody with a simple chordal accompaniment below it, with the chords occuring at the same time as melody notes. The four algorithms are as follows; the effect of each algorithm is illustrated in Figure 4.

- In algorithm one, all musical information is combined into one stream of events. Whenever a note starts, it chooses the top note of any notes that start at the same time. This will result in many overlapping notes, as a note that is sustained in one track will cover the start of other notes. For the purpose of evaluation, note lengths were truncated until the result was monophonic. As can be seen in Figure 4b, extra notes are often included in the melody.

- Algorithm two makes use of the structure of MIDI files (tracks and channels) by processing each channel separately; for each channel the information is processed as in algorithm one above. The channel with the highest average pitch is then chosen as the melody. The algorithm, illustrated in Figure 4c, works well for the example but in practice sometimes wrongly identifies a high accompanying part.

- In algorithm three the top notes for each channel are chosen as in algorithms one and two, then the channel with the highest first-order predictive entropy is selected. In the example the higher part had the greatest entropy and so was chosen, as in Figure 4d, instead of the melody.

- Algorithm four uses heuristics to split each channel into parts. It does so by allocating successive notes to parts that have the closest proximity in pitch to the current note in the stream of notes. A new part is only created if the current note occurs at the same time as the existing parts. The part with the highest entropy is then chosen as the melody. The splitting algorithm is a little simplistic as it only compares the current note to previous notes. As can be seen in Figure 4e, the part selected goes from middle C to F, instead of back to middle C, because in the note stream the F occurred before the second middle C. If the simultaneous notes C and F were reversed in the note stream a rather different part would be produced.

From our analysis of the needs of music databases users and from well-known results in music perception, we believe that each of these algorithms has potential as a method of extracting monophonic melodies from music. However, application of these methods to actual music shows that they can produce very different results. We applied the four algorithms to ten MIDI files, representing a range of styles and methods of organisation within the file. Each of our eight listeners were presented with the original MIDI file and the four automatically-extracted melodies for each of the ten files.[1]

The melodies for each piece of music were presented in random order. The listeners were able to play the pieces as often as they wished and sometimes only played a small portion of the melodies if that was sufficient to make a decision. The melodies were ranked from one to four according to how well they represented the melody of the original piece. The listeners were asked to consider the presence of extra notes and absence of melody notes in their evaluation, and were permitted to give melodies equal ranking where appropriate.

Results

The sum of the ratings for each algorithm for each piece are shown in the table at Table 1. Low scores indicate a high rating. The range of possible scores is 8 to 32. As can be seen from the results, most algorithms performed well for some pieces and badly for others. The only algorithm to work consistently well was algorithm one—the most naive of the methods we considered. Interestingly, for pieces 5 and 8 the first three algorithms produced identical melodies but were ranked somewhat differently. For piece number 5 ("Scotland the Brave") it is the authors' opinion that algorithm four performed the best, as accompanying notes at the start and during the piece were removed. This difference was not detected by the evaluators, however.

Sometimes music is perceived differently if it is recognised. In this experiment, however, the algorithm rankings were similar regardless of whether or not the music was recognised. The music experience of the evaluators did not reveal any clear cut trends in algorithm preference. The least skilled group (three people) and the most skilled group (three people) ranked algorithm one as the best whereas the middle group (two people) ranked algorithm three the best overall.

In conclusion, choosing the top notes usually best selects the melody of a piece of music, but garners extra notes. The other three methods often select a part which has no notes in common with the melody of the music; as our results show, there can be almost no notes in common between the melodies generated by these algorithms.

There are several improvements that can be made to the above algorithms to make them more effective. The first algorithm often includes notes that have a much lower or higher pitch than the majority of the notes. In some circumstances these could be removed. The part extraction algorithm could be improved to consider all notes that start at the same time as a group. Entropy could be combined with average pitch when selecting the melody part. We are currently exploring these approaches.

[1] These MIDI files are available from **http://www.mds.rmit.edu.au/~sandra/melexp**.

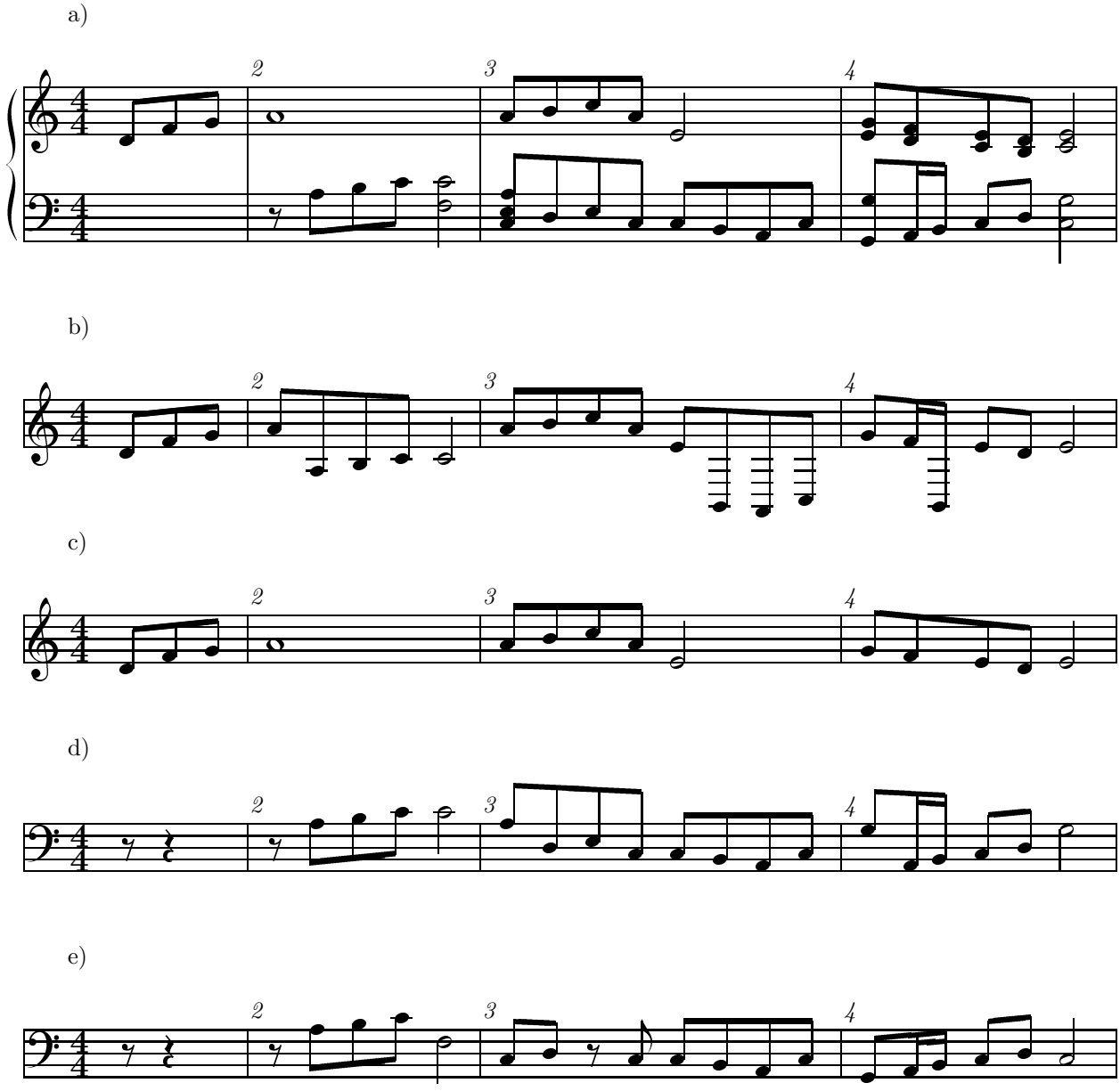

Figure 4: *A short piece of music (a) and the resulting monophonic "melodies" (b–e) extracted by each of the four algorithms.*

	Algorithm				Best
	One	Two	Three	Four	Alg
1	18, 0.70/0.13	30, 0.00/0.00	14, 1.00/1.00	17, 0.76/1.00	3
2	10, 1.00/1.00	24, 0.00/0.03	19, 0.16/0.47	21, 0.16,0.47	1
3	9, 1.00/1.00	32, 0.00/0.00	20, 0.13/0.19	19, 0.13/0.19	1
4	14, 1.00/1.00	14, 1.00/1.00	13, 1.00/1.00	32, 0.35/0.55	3
5	11, 1.00/1.00	22, 1.00/1.00	13, 1.00/1.00	13, 0.93/1.00	1
6	16, 0.99/0.34	12, 1.00/1.00	16, 0.91/0.38	32, 0.00/0.00	2
7	14, 0.99/0.69	12, 1.00/1.00	26, 0.00/0.00	24, 0.00/0.00	2
8	22, 1.00/1.00	14, 1.00/1.00	16, 1.00/1.00	25, 0.43/0.93	2
9	13, 1.00/1.00	31, 0.11/0.59	21, 0.02/0.03	20, 0.02/0.03	1
10	13, 1.00/1.00	14, 0.78/1.00	32, 0.22/0.25	19, 0.30/0.49	1

Table 1: *Results of ranking melody extraction algorithms. The first number in each algorithm column is the sum of the rankings given for the melodies by the 10 evaluators. The second number is the proportion of notes in common with the melody selected as the best by the evaluators expressed as a ratio of number of common notes over the number of notes in the best melody. The third number is the ratio of the number of common notes over the number of notes in the melody generated by the algorithm. For example, the most successful algorithm for piece number one was judged to be algorithm number 3. The number of notes it had in common with algorithm one divided by the number of notes in algorithm three's melody equals 0.70. The number of common notes divided by the number of notes in algorithm one's melody is 0.13.*

6 Searching Music

Several researchers have explored the problem of searching music databases for melodies. Eaglestone [9] proposed extending the relational model for the storage of music. He made use of temporal database techniques to retrieve music. Hawley [12] performed exact matches using grep on melodies that were stored as a series of relative pitches in a text format.

Chou et al. [6] used Pat trees to index melodies. The notes of each measure of the melody were combined to determine the "chord" and this chord sequence was used as an index term. The music was stored in a diatonic fashion, not allowing for notes that are outside the key of the piece of music.

Kageyama et al. [14] created indexes of melodies, indexing from the start of each phrase, reasoning that users will usually start a query at the start of a phrase. This reduces the number of terms required to index each melody at the expense of supporting queries that start in the middle of phrases. Inexact matches were permitted. They refer also to work by Yamamoto, who developed a system allowing exact matching only.

In each of these proposed techniques, and in other work on indexing music [1, 2, 17, 18, 20], only indexing of monophonic melodies was considered. Each of these techniques could be used in conjunction with a melody extraction algorithm.

There are several factors that must be considered when designing search techniques for music. A music query will not necessarily begin at the start of a melody; it may, however, begin at the start of a musical phrase, as suggested by Kageyama et al. [14]. Queries may possibly be chordal or may even cross from one musical part to another. It may indeed be a part of the melody that is being queried, or a part of the accompaniment.

To allow for queries that start anywhere within a melody, indexing via n-grams could be useful, where an n-gram is a sequence of n consecutive notes and each such sequence is extracted from the music; similar techniques are used for the related problems of string indexing and genomic indexing [22, 23]. With melodic data, however, it is not particularly useful to represent the data as an absolute pitch, since the same melody can be played or sung at different pitches. Instead a relative pitch should be used, that is, the interval or distance between adjacent notes.

Another approach used for representing melodies is to use contour instead of precise intervals. A melody's contour describes a melody's shape in terms of pitch direction only; a melody can then be represented with an alphabet of three symbols, say "u" for up, "d" for down and "s" for same pitch. For long strings of contour information, it may be possible to uniquely identify a melody in a database. The advantage of using contour is that if a query is entered by humming the contour is usually correct, but the pitch intervals may not be, due to the difficulty in singing accurately. In a large database, however, more precise information may be needed.

Another possible method of representing the melody is to store the relative pitches reduced to the scope of one octave, that is, as modulo-12 intervals. This may be of use from a musicological perspective, or when looking at the harmony of a musical work, but it loses information about the aspect of melody that people remember the best—the contour. If used, it would need to be combined with a search on contour.

We examined the distribution of the contour n-grams found in a collection of 500 MIDI files. These files contained a total of 2 697 tracks that were each processed individually to extract their melody n-grams. A typical query of six notes would result in about 330 tracks being returned. In the collection of nearly 15 000 files, about 1 220 tracks of about 65 000 are retrieved. Assuming that the user needs to listen to a 10-second excerpt of each melody, it would take over three hours to determine which answers are relevant. Clearly a longer string of notes would be required in queries, or greater precision in the search for melodies. To retrieve less than 10 tracks from 2 697 using contour requires a query consisting of 12 notes.

The data analysis described above suggests that using melody contour as the only basis for preparing answers to a query is not sufficient. In spite of this, it may still be useful to match on contour. Once a collection of potential answers is identified using contour, they can be examined more closely, using exact interval or modulo-12 interval n-grams. If modulo-12 is used, then the retrieved melodies will need to be examined for contour. That is, we believe that a combination of approaches will be required.

In work on contour Ghias et al. [11] used a collection of MIDI files as data. Melodies were extracted from these files as contour strings. These were stored in a text file and searched for matches to users' hummed queries. McNab [17, 18] converted melodies to contour strings, but also explored the effect of storing exact interval and contour and exact rhythm information. He discovered that after exact interval and rhythm, exact contour and rhythm was the combination with the greatest discriminatory power. He used a modified version of the edit distance algorithm to allow for two dimensions: pitch and duration.

Other methods explored include the encoding of whether a note is stressed or not [1, 2, 20], which would be important for ranking similar melodies, since experience suggests that those melodies that only differ on an unstressed note are more similar than those that differ on a stressed note. The approach of both Schaffrath [20] and Bakhmutova et al. [1, 2] was to use diatonic information, that is, the notes were numbered according to the note number in the scale or key of the music.

7 Conclusions

Much music is now available online, as either performance information or music notation. The value of this music would be greatly enhanced by the existence of techniques for searching it for familiar melodies. A central problem in such searching is that stored music typically consists of multiple simultaneous parts, each of which can be chordal, and it is often the case that none of these parts has a simple correspondence to the perceived melody.

We have reviewed the needs of likely users of music databases and the psychology of music perception, and based on this review proposed several techniques for extracting likely monophonic melodies from polyphonic music. Our experiments with ten pieces of music and eight subjects indicate that the simplest technique—in which the extracted melody consists of highest-pitch notes appearing in any track—is the most effective, despite the fact that it often garners additional notes.

We have also outlined possible approaches to techniques for searching of music databases, which

we expect to be based on a combination of relative pitch, contour, and possibly stress. That is, we expect that successful matching will rely on evidence from a variety of aspects of the music. These matching techniques, like previous matching techniques described in the literature, rely on monophonic melodies: extraction of such melodies is a necessary precursor to searching of music.

We are now planning further experiments, to complete the design of algorithms for resolving melody queries on collections of music. We will use automatically-extracted melodies for melody matching on a large collection of MIDI files, then use relevance assessment to judge the accuracy of the retrieval process. These experiments will provide further exploration of both melody extraction techniques and music indexing techniques.

Acknowledgements

We thank John Harnett for invaluable sys-admin assistance.

References

[1] I.V. Bakhmutova, V. D. Gusev, and T. N. Titkova. A search and classification of imperfect repetitions in song melodies. *Acta et Commentationes Universitatis Tartuensis: Quantitative Linguistics and Automatic Text Analysis*, 827:20–32, 1988.

[2] I.V. Bakhmutova, V.D. Gusev, and T.N. Titkova. The search for adaptations in song melodies. *Computer Music Journal*, 21(1):58–67, Spring 1997.

[3] J.C. Bartlett and W.J. Dowling. Recognition of transposed melodies: A key-distance effect in developmental perspective. *Journal of Experimental Psychology: Human Perception and Performance*, 6(3):501–515, 1980.

[4] D.C. Blair. STAIRS redux: Thoughts on the STAIRS evaluation, ten years after. *Journal of the American Society for Information Science*, 47:4–22, 1996.

[5] H. Charnasse and B. Stepien. Automatic transcription of german lute tablatures: an artificial intelligence application. In *Computer Representations and Models in Music*, pages 143–170. Academic Press, 1992.

[6] T.-C. Chou, A.L.P. Chen, and C.-C. Liu. Music databases: Indexing techniques and implementation. In *Proceedings IEEE International Workshop in Multimedia DBMS*, 1996.

[7] D. Deutsch. Grouping mechanisms in music. In *The Psychology of Music*, chapter 4, pages 99–134. Academic Press, Inc., 1982.

[8] W.J. Dowling. Scale and contour: Two components of a theory of memory for melodies. *Psychological Review*, 85(4):341–354, 1978.

[9] B.M. Eaglestone. Extending the relational database model for computer music research. In A. M. and A. Pople, editors, *Computer Representations and Models in Music*, pages 41–66. Academic Press, 1992.

[10] R. Frances. *La Perception de la Musique*. 1958. Translated by W.J. Dowling.

[11] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming - musical information retrieval in an audio database. In *ACM Multimedia 95 - Electronic Proceedings*, 1995.

[12] M. Hawley. Structure out of sound. Email discussion with the author about his thesis.

[13] International Standards Organization. *Standard Music Description Language (SMDL) ISO/IEC DIS 10743*. Draft International Standard.

[14] T. Kageyama, K. Mochizuki, and Y. Takashima. Melody retrieval with humming. In *ICMC Proceedings 1993*, 1993.

[15] C.L. Krumhansl and R.N. Shepard. Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of Experimental Psychology: Human Perception and Performance*, 5(4):579–594, 1979.

[16] A. Marsden. Modelling the perception of musical voices: a case study in rule-based systems. In *Computer Representations and Models in Music*, pages 239–263. Academic Press, 1992.

[17] R.J. McNab. Interactive applications of music transcription. Master's thesis, Department of Computer Science, University of Waikato, New Zealand, 1996.

[18] R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson, and S.J. Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Digital Libraries Conference*, 1996.

[19] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[20] H. Schaffrath. The retrieval of monophonic melodies and their variants: Concepts and strategies for computer-aided analysis. In A. Marsden and A. Pople, editors, *Computer Representations and Models in Music*, pages 95–110. Academic Press, 1992.

[21] R. van Egmond and D.-J. Povel. Perceived similarity of exact and inexact transpositions. *Acta Psychologica*, 92:283–295, 1996.

[22] H. Williams and J. Zobel. Indexing nucleotide databases for fast query evaluation. In *Proceedings of Advances in Database Technology (EDBT'96)*, pages 275–288, Avignon, France, March 1996.

[23] J. Zobel and P. Dart. Finding approximate matches in large lexicons. *Software—Practice and Experience*, 25(3):331–345, March 1995.