



Thank you for downloading this document from the RMIT Research Repository.

The RMIT Research Repository is an open access database showcasing the research outputs of RMIT University researchers.

RMIT Research Repository: <http://researchbank.rmit.edu.au/>

Citation:

Uitdenbogerd, A and Zobel, J 2002, 'Music ranking techniques evaluated', in M. Oudshoorn (ed.) Proceedings of the 25th Australasian Computer Science Conference, Sydney, Australia, 28 January - 1 February 2002, pp. 275-283.

See this record in the RMIT Research Repository at:

<https://researchbank.rmit.edu.au/view/rmit:30949>

Version: Published Version

Copyright Statement: Copyright c2001, Australian Computer Society, Inc.

Link to Published Version:

<http://crpit.com/confpapers/CRPITV4Uitdenbogerd.pdf>

PLEASE DO NOT REMOVE THIS PAGE

Music Ranking Techniques Evaluated

Alexandra L. Uitdenbogerd

Justin Zobel

School of Computer Science & Information Technology, RMIT University
GPO Box 2476V, Melbourne 3001, Australia
{alu,jz}@cs.rmit.edu.au

Abstract

In a music retrieval system, a user presents a piece of music as a query and the system must identify from a corpus of performances other pieces with a similar melody. Several techniques have been proposed for matching such queries to stored music. In previous work, we found that local alignment, a technique derived from bioinformatics, was more effective than the n-gram methods derived from information retrieval; other researchers have reported success with n-grams, but have not compared against local alignment. In this paper we explore a broader range of n-gram techniques, and test them with both manual queries and queries automatically extracted from MIDI files. Our experiments show that n-gram matching techniques can be as effective as local alignment; one highly effective technique is to simply count the number of n-grams in common between the query and the stored piece of music. N-grams are particularly effective for short queries and manual queries, while local alignment is superior for automatic queries.

Keywords: music matching techniques, n-gram matching, edit distances, manual and automatic queries

1 Introduction

A music retrieval system finds pieces of music that are similar to a user-provided query. For example, in current systems a user can sing a fragment of a melody, or play it at a keyboard. The system matches this query against a corpus of pieces in a note-based format such as MIDI files. The intention of such systems is to allow a user to search for a piece with a particular melody, to find alternative arrangements of a given composition, or to check that a new composition is indeed original.

Stored music can be queried using a variety of techniques derived from information retrieval. Techniques for matching music can differ in several ways: the representation of the music can be based on exact pitch, relative pitch, pitch contour, note stress, rhythm, or a combination of these [1, 2, 3, 8, 10]; and the matching of query to this music representation (or *melody string*) can be based on n-grams or dynamic programming [5, 7, 9, 11]. However, for matching to be accurate, significant problems must be addressed. For example, most arrangements are polyphonic, with chords and multiple instruments in use simultaneously, but only some of the notes contribute to the melody. As the melody can shift between instruments, and as ornamentation, counter-melodies, and so on, may be present, identification of

the melody is a difficult task.

In previous work we reported that local alignment effectively matches melody strings [16], particularly in comparison to n-gram techniques. These experiments, part of the ongoing MIRT project at RMIT [14], also compared a range of techniques for extracting melody strings from MIDI files. However, our experiments had significant limitations. One was that we did not test other ranking formulations based on n-grams. Another was that we used as queries a set of melody strings automatically extracted from MIDI files, for which the relevant pieces of music were all the files that we could locate in our corpus that were performances of the same piece of music. These files are polyphonic and thus the melody extraction technique is subject to error; however, such *automatic* queries are clearly one of the practical uses that can be made of music corpuses.

In this paper we address these limitations. First, we have used a musician to create a set of 30 *manual* queries, based on 30 of the MIDI-file queries used in the earlier experiments. These queries therefore have the same set of relevant pieces; in other work [13] we report on the process of gathering human relevance judgements. Second, we have evaluated a wider range of n-gram matching techniques. Using the same set of melody string extraction techniques as previously, we show that simple n-gram formulations can work very well indeed, particularly for short queries and manual queries. Downie [5] has reported successful experiments with n-gram matching based on classic information retrieval similarity techniques, in which a “TF-IDF” formulation was used to rank melody strings using both overall n-gram rareness in the corpus and the frequency of occurrence of the n-grams in each melody string. Our experiments reported here, using a more realistic collection, show that TF-IDF ranking is not as effective as either local alignment or simple coordinate matching with n-grams. We show that, overall, matching 5-grams from the query against each channel of stored pieces provides good effectiveness for all query types.

Interestingly, matching techniques for automatic queries are not necessarily effective for manual queries, and melody string extraction techniques that allow good matching of automatic queries can be poor for matching of manual queries. In an ideal system, the different kinds of queries would be supported by different query evaluation mechanisms.

2 Melody Matching

In earlier work [16], we proposed a three-stage approach to melody matching, consisting of melody extraction, melody standardisation, and similarity measurement.

Melody extraction is based on the assumption that queries are fragments of melodies, for example a



Figure 1: *Example melody fragment that contains a leap of more than an octave (from Domine Deus by Mozart, K427).*

phrase. Based on results of unpublished work, we conclude that it is necessary to reduce the set of notes against which the query should be matched to just the melody of each piece of music, as there would be far too many irrelevant matches otherwise. Using melody extraction also reduces the costs of melody matching. In our earlier work [15] we implemented four melody extraction techniques and evaluated these by asking volunteers to rank the extracted melodies in terms of similarity to the melody of the original piece of music. The methods were based on several results from music perception: listeners will usually hear the part with the highest pitch as the melody unless it is monotonous [6]; parts within music are grouped based on proximity of pitch [4]. The melody extraction algorithms tested were:

All-mono: combine all musical parts and include the highest pitch note starting at any instance as part of the melody. This technique includes many extra notes.

Entropy-channel: use the all-mono technique for each individual channel, then select the resulting melody that has the highest first-order entropy.

Entropy-part: split the music into parts using proximity and time information, then select the resulting melody that has the highest first-order entropy.

Top-channel: use the all-mono technique for each individual channel, then select the resulting melody that has the highest average pitch.

The all-mono approach was judged by listeners to be the most effective melody extraction technique despite the extra irrelevant notes found in many of the generated melodies.

In subsequent work we tested the different extraction techniques in the context of music retrieval, using automatic queries and the database discussed below. We additionally tested the technique of *all-channels*, in which each piece is represented multiple times, by a melody extracted from each of its separate channels. These results found that local alignment, and the all-mono melody extraction method gave the greatest effectiveness, and that n-gram frequency measures, contour melody representation and the longest common subsequence matching technique were poor.

Melody standardisation involves encoding just those aspects of the melody that are to be used for matching. For example, we can encode the intervals between melody notes to allow melodies to be matched regardless of the key of the query or pieces in the database. It is important to not remove too much information in this standardisation process, for example by only encoding contour (in which the only values represented are up, down, and same), or there will be too many irrelevant matches.

In our previous and current experiments, we have considered three standardisation techniques: melody contour, directed modulo-12 intervals, and exact intervals. We illustrate the three methods with an example, the notation of which is shown in figure 1. The contour representation of this melody fragment, using U for up, D for down and S for same pitch, is:

SSUDDDDDD

Using exact interval representation we use the number of semitones between successive notes:

0 0 16 -4 -3 -5 -4 -1 -2.

Directed modulo-12 intervals retain the direction information but reduce any intervals greater than an octave to the harmonically similar interval that is no more than an octave in size:

0 0 4 -4 -3 -5 -4 -1 -2.

Contour representation has the advantage that singers usually get the contour of a melody right but usually don't sing the intervals accurately, an important consideration when queries are sung. A melody query would need to be quite long for relevant answers to be found, however. Exact interval representation provides for more accurate matching and the directed modulo-12 interval method allows a smaller representation with little loss in accuracy. All three methods have the problem that an error in a pitch can result in two consecutive errors in a matched melody when using string-based pattern matching techniques. For example, if a query melody contained a G instead of an F in bar 3, its representation would be:

0 0 16 -4 -3 -5 -2 -3 -2.

which has two adjacent symbols that differ from the original melody despite only one note being different.

The final stage of melody matching is *similarity measurement*, the calculation of a statistic representing how similar the query is to a piece of music. Similarity measurement is applied to the standardised representations of the melodies that are to be compared. We have experimented with various dynamic-programming-based techniques and n-gram-based techniques. In our previous experiments, the dynamic programming (or "edit distance") techniques tested were local alignment and longest common subsequence. For experiments not reported here we also implemented a variation of longest common substring.

In its simplest form, dynamic programming involves creating a two-dimensional array, the dimensions of which are the length of the pattern (query) by the length of the text to which it is being compared. The array is filled according to a set of conditions. Below is the definition used for local alignment in our experiments:

Assume a represents the array, q and p represent the query melody string and piece to match against respectively, and array index i ranges from 0 to query length and index j ranges from 0 to piece length:

$$a[i, j] = \max \begin{cases} a[i-1, j] + d & (i \geq 1) \\ a[i, j-1] + d & (j \geq 1) \\ a[i-1, j-1] + e & (q(i) = p(j) \\ & \text{and } i, j \geq 1) \\ a[i-1, j-1] + m & (q(i) \neq p(j)) \\ 0 & \end{cases}$$

where d is the cost of an insert or delete, e is the value of an exact match, and m is the cost of a mismatch.

These techniques can be varied by choosing different penalties for insertions, deletions, and replacements, and different scores for correct match.

For longest common subsequence, array elements are incremented if the current cell has a match, otherwise they are set to the same value as the value in the upper left diagonal. That is, indels and mismatches

do not change the score of the match, having a cost of zero.

The n-gram techniques involve counting the common (or different) n-grams of the query and melody to arrive at a score representing their similarity. Our first n-gram method was a simple sum of frequencies of n-grams that were common to both the query and the melody to which it is being compared. The second was the Ukkonen n-gram measure, which is based on the difference in n-gram frequencies between the query and the melody. In our previous experiment these methods were tried with $n = 4$. The sum of frequencies method was normalised for piece length, while the Ukkonen measure was not.

In the current work we explore different n-gram lengths and normalisation techniques. In addition we test the effect of ignoring term (n-gram) frequency information, and simply counting the distinct n-grams that occur in both the melody and the query. This technique is known as “coordinate matching” in the literature. We also test the TF-IDF technique that is widely used in text retrieval [17] and used by Downie [5] for music retrieval. TF-IDF is where the similarity assigned to a piece of music is the sum of the weights of the n-grams in the piece that match n-grams in the query. The weight of an n-gram is determined by its frequency in the piece and by the reciprocal (or log of the reciprocal) of the number of pieces containing it.

As an example, consider 3-grams and the contour representation of the previous melody example, SSUDDDDDD, and compare it to a slightly different contour string, SSUDDDUDDD. The “sum common” method gives a score of 5, coordinate matching gives a score of 4, and the Ukkonen measure gives a score of 5. Using some IDF figures based on the *all-mono* melody collection, the TF-IDF score for the pair is 7.69.

3 Previous Results and Current Motivation

In our previous work we compared various similarity measurement and melody standardisation methods. The test database consisted of a collection of MIDI files downloaded from the internet. The melody extraction algorithms described earlier were used to produce a set of melodies to which query melodies could be compared. Query melodies were automatically generated using the same algorithms. The answers retrieved were considered relevant if they were one of a collection of MIDI files that were a version of the same piece of music as the original query.

The results clearly indicated that local alignment was an effective method for similarity measurement, while *all-mono* was the best of the melody extraction methods. The n-gram methods used were generally poor, as was the longest common subsequence method.

A more detailed analysis of the experimental results for 30-note queries using modulo-12 intervals revealed that:

- N-gram counting was always worse than local alignment, but tended to follow the same pattern of success and failure as local alignment. For example, if n-gram counting was successful for a particular query, local alignment was also successful for that query.
- N-gram counting tended to favour long pieces of music with high repetition.
- The Ukkonen measure seemed to include many very short files in its answers. It was the only method that didn't use length normalisation in

our original experiment. Re-running queries using the Ukkonen measure with normalisation improved the recall-precision slightly but it was still worse than the other methods trialled in the original experiment.

- Certain queries were always unsuccessful in retrieving answers other than the piece from which they were extracted. This usually occurred when the melody was not cleanly extracted, that is, the *all-mono* method produced a melody with many accompanying notes in it. These same queries typically caused the other melody extraction methods to select a non-melody part.

However, n-grams remain an attractive matching technique. In genomics, for example, they can be used for an initial, rapid coarse search, allowing a subsequent more careful search of a smaller set of potential answers. Thus it is important to determine if there is a better n-gram-based method for finding answers. The experiments described below show the results of our exploration of this problem.

Another issue is whether these results, determined with automatic queries, were also valid for manual queries. We gathered a set of manual queries by asking a musician to listen to the MIDI files of pieces which had been selected previously as the basis for automatic queries, and to play a melodic query representing the piece. This allowed us to use the same relevance judgements as before. Using these queries we re-evaluated the extraction, standardisation, and matching techniques.

4 Experiments

We had two aims in these experiments: to comprehensively evaluate n-gram methods and to explore whether melody matching techniques for automatic queries were valid for manual queries. These experiments are in three parts: evaluation of n-gram methods; comparison of the best n-gram methods to local alignment methods; and comparison of results on manual queries.

Experiment 1

A set of 28 queries was compiled by manually locating pieces of music for which there were at least two distinct versions in the collection. Melodies were extracted from these using the melody extraction techniques listed above. These were presented as the *automatic* queries to the music databases. The music databases consisted of automatically extracted melodies for each of the musical works in the MIDI file collection that we had downloaded from the Internet. The average number of relevant matches per query was about 4.5.

For each query we used three different n-gram counting methods:

- c:** Count distinct occurrences in the melody of each n-gram that occurs in the melody. (This yields a maximum number equal to the query length minus the n-gram length plus 1). The technique is also known as “coordinate matching” in the literature.
- n:** Sum the frequency of occurrence in the melody of n-grams that also occur in the melody. This is the technique used in our earlier experiments.
- u:** The Ukkonen measure: sum the frequencies of n-grams that occur in one but not the other.

The counting methods were varied by use of different length normalisation techniques, including no

Table 1: *Eleven-point precision averages for ngrams with n=5. The automatic queries were processed with the same extraction technique as the melody database. The second column shows the normalisation amount used, with l indicating division by the log of the length and the others being the k-th root of the length. Query lengths used are 8, 20 and 30. Databases are: all-mono (amdb), entropy-channel (ecdb), and top-channel (tcdb).*

		c			n			u		
		008	020	030	008	020	030	008	020	030
amdb	0	29.81	48.10	46.71	30.47	36.73	35.80	00.04	00.04	00.05
	1	29.37	42.59	38.75	30.61	42.28	40.29	09.48	43.22	42.62
	9	35.04	48.93	48.25	31.00	39.16	36.67	00.04	00.04	00.05
	l	35.04	48.93	48.25	31.00	39.27	36.40	00.04	00.04	00.05
ecdb	0	24.51	38.59	38.34	21.65	28.45	22.17	00.12	00.49	00.71
	1	23.69	29.92	24.72	24.61	27.53	21.11	04.19	35.12	37.00
	9	26.24	38.35	38.15	22.27	27.78	22.73	00.13	00.49	00.78
	l	26.24	38.36	38.15	22.27	27.64	23.22	00.13	00.67	03.52
tcdb	0	23.21	34.07	33.65	24.02	30.94	30.17	00.02	02.28	02.61
	1	19.07	23.52	23.16	23.02	26.16	25.14	09.50	32.75	33.60
	9	26.20	33.78	33.78	24.40	31.01	30.56	00.02	02.61	02.64
	l	26.20	33.59	34.04	24.45	31.50	30.42	00.03	02.39	05.64

normalisation, division by the length of the melody, division by the square root, cube root and ninth root of the length, and the more typical division by the log of the length. These length normalisation techniques have very different effects: the ninth-root method, for example, distinguishes amongst very short pieces but otherwise has little effect, whereas some of the other methods distinguish amongst pieces of all lengths.

Query lengths of 8, 20, and 30 were tried on all n-gram lengths from 3 to 8. All query processing used the modulo-12 with direction method of melody standardisation. Query melodies used the same extraction method as those of the database against which they are being compared. We also tried matching query melodies against the *all-channels* database, which contains the extracted melody of each channel in each melody, using the top notes extraction method.

An answer was considered relevant if it was one of the versions of the query piece that were located by examining files with likely file names. Methods were evaluated by calculating eleven-point precision averages and precision at 20 documents retrieved, but the results for the measurement techniques are completely consistent and we only report the former.

Results comparing counting methods, normalisation, and melody extraction technique are shown in Tables 1 and 2. The former shows results when query and database have the same melody extraction technique, the latter for different melody extract techniques against the *all-channels* database. *Entropy-part* results were so poor in all experiments that we have chosen not to report them; *top-channel* results are excluded from tables when they too were poor.

These results illustrate the difficulty of choosing a “best” matching technique. The Ukkonen (u) method is generally poor, but for long queries gives acceptable performance for one kind of normalisation. (Due to these results, however, particularly on short queries, we do not consider it further.) Overall, coordinate matching has been the most effective method, working well on the *all-mono* database and with *all-mono* and *entropy-channel* melody extraction against the *all-channels* database. It rivals the local alignment method tested in our earlier paper [16]. Also of note is that the best normalisation methods seem to be

those that only affect the scores by a small amount, namely log normalisation, dividing by the ninth root of the melody length and no normalisation.

The very best results were achieved by the entropy-channel method using the count-distinct n-gram measure. Other observations are that longer query lengths again lead to more relevant answers being retrieved when using count-distinct or Ukkonen measures but not when using the sum-of-frequency measure. This promising result for counting distinct n-grams suggests that it is possible to successfully produce a useful n-gram-based index for melody retrieval. It also makes clear that term frequency is unimportant in melody retrieval.

Results exploring variation in n-gram length are reported in Tables 3 and 4, for the count-distinct or coordinate-matching method and several length normalisation techniques. These results show that $n = 5$ to $n = 7$ gives best results. We report only $n = 5$ for the remainder of our experiments.

Experiment 2

We then evaluated TF-IDF ranking and local alignment, using the same framework of query lengths, melody extraction techniques, and kinds of length normalisation. These are notated as:

- a**: Local alignment.
- i**: TF-IDF ranking using unmodified reciprocal for inverse frequency.
- L**: TF-IDF ranking using log of reciprocal.

Results are shown in Tables 5 and 6. The local alignment results correspond to those of our earlier experiments. TF-IDF is somewhat weaker than the other methods, while local alignment is comparable to the best n-gram methods for both short and long queries. However, local alignment is highly sensitive to the normalisation method: with strong normalisation, the scores are meaningless and almost no relevant pieces are found with this collection. These results are startlingly poor, but are consistent with trends observed as normalisation is weakened, and thus are not, for example, the result of experimental

Table 2: *Eleven-point precision averages for ngrams with n=5. Twenty-eight automatically extracted queries against the all channels melody database. (run m3d12s28)*

		c			n			u		
		008	020	030	008	020	030	008	020	030
amdb	0	23.66	39.59	43.51	23.06	29.92	26.36	00.05	00.07	00.38
	1	15.13	22.34	18.25	20.56	24.90	23.25	03.91	36.18	33.77
	9	25.87	43.45	46.67	22.09	31.23	26.81	00.18	00.14	00.49
	1	25.76	42.00	47.24	22.11	31.26	26.69	00.13	01.01	05.87
ecdb	0	27.57	46.71	48.73	21.30	29.13	24.16	00.09	00.42	00.74
	1	16.19	21.57	17.73	19.98	22.02	18.68	03.15	36.18	42.39
	9	27.34	47.88	46.76	21.11	29.63	24.05	00.21	00.47	00.82
	1	26.98	44.29	46.39	21.22	29.95	23.99	00.18	00.71	03.59
tcdb	0	24.99	42.11	41.02	21.57	30.67	27.49	00.09	01.86	02.50
	1	16.39	23.49	25.11	21.87	25.20	24.68	07.75	39.72	37.95
	9	28.08	43.63	43.72	23.18	30.69	27.68	00.21	02.33	02.68
	1	27.57	42.39	42.13	23.24	31.18	28.97	00.18	02.71	05.97

error. We ran similar experiments using a different query set, of 46 automatic queries gathered with the same methodology, and observed similar results.

Previous published work [5] suggests that the TF-IDF technique of information retrieval works well for melody retrieval. Our results indicate, however, that it would not perform as well as does ignoring term frequency. Another technique often discussed [17] uses the log of the term frequency instead of the term frequency itself, to reduce the weighting of terms that are repeated. We predict that this also would perform less well than ignoring the term frequency.

Experiment 3

Our final experiment was to produce comparable results for a set of 30 manual queries, created by a volunteer who listened to each piece in turn and created a melody query by playing on a synthesizer keyboard connected via MIDI to a sequencing program. There was overlap in the pieces represented in the manual set of queries and the other two sets. Where this occurred, the same MIDI file was used as the basis of the query. We restricted our experiment to log normalisation and modulo-12 melody standardisation, which were shown above to work well.

Table 7 shows the results for manual queries. The results when using full manual queries, as opposed to queries truncated to 8, 20, or 30 notes, are shown in 8. Coordinate matching is clearly superior to all other methods when applied to shortened manual queries. For longer queries and the automatically extracted queries, however, local alignment performed best. This means that when exact matches are to be found local alignment works well but when there are minor differences, the n-gram approach succeeds and the local alignment approach fails. In both cases the shortest queries are most successfully handled by coordinate matching.

Perhaps the most startling aspect of these results is that only the *all-channels* database has produced acceptable results. The best effectiveness with *all-channels* is over 40; the best with the other methods is less than 19.

It can be clearly seen that the best n-gram length is five. We hypothesised that part of the success of the 5-gram coordinate matching technique is that it

screens out melodies with short matches. If this were so, then a technique that only allowed matches of a minimum length of five should perform well. We are currently exploring these possibilities.

In further tests we measured the performance of contour melody extraction with n-grams and local alignment, and found it extremely poor. The best results ranged from 0.47 for 8-note queries to 15.03 for 30-note queries.

Analysis

On close analysis of the best n-gram and dynamic programming techniques for modulo-12 intervals, we observed that all relevant answers retrieved within the top twenty by the local alignment technique were also retrieved by n-gram coordinate matching. The relevant answers retrieved by coordinate matching but not by the dynamic programming approach tended to have significant sections of the query matching widely spaced sections of the answer. Upon examination of these answers, the reason seems to be that the query melody partially matches different variations of the same theme within the answer melody. The coordinate matching n-gram method allows these partial matches to be accumulated, without including repeated fragments. We have already seen that when repetition is included, the precision of answers is reduced.

Another interesting observation was that the number of distinct n-grams of length five for the *all-mono* approach was 50% greater than that for *all-channels*. The *entropy-channel* approach had approximately 60% of the n-grams of *all-channels*. This suggests that there is little penalty in using the *all-channels* approach for melody indexing.

5 Conclusions

We have tested a wide range of similarity measures for melody matching, using both automatic and manual queries. Our experiments have revealed that the best approach to using n-grams for ranking melodies is to ignore term frequency completely and have grams of length five. When used in this manner, the technique rivals local alignment in its ability to produce relevant

Table 3: *Eleven-point precision averages for different n-gram lengths using the 28 automatic melody queries and the “count distinct” measure. Column two shows the n-gram length. The headings 0, 9, and l refer to the type of normalisation applied to the scores. Same melody extraction technique for query and database.*

		0			9			l		
		008	020	030	008	020	030	008	020	030
amdb	3	13.17	27.74	27.54	21.68	33.42	32.85	20.12	34.17	35.12
	4	24.06	38.15	40.41	25.89	43.40	46.25	25.53	41.80	47.44
	5	23.66	39.59	43.51	25.87	43.45	46.67	25.76	42.00	47.24
	6	17.87	43.48	46.93	17.67	43.71	47.91	17.67	42.14	47.23
	7	16.77	40.80	43.26	16.02	38.95	42.83	16.02	38.95	42.73
	8	11.96	37.72	41.84	11.85	36.32	41.60	11.85	36.32	41.60
ecdb	3	14.84	36.92	39.28	20.06	39.42	43.59	17.76	39.03	41.02
	4	26.35	46.16	46.99	27.47	47.14	49.00	25.65	42.48	46.23
	5	27.57	46.71	48.73	27.34	47.88	46.76	26.98	44.29	46.39
	6	22.96	48.93	49.28	22.27	47.49	47.16	22.27	47.79	46.74
	7	17.18	49.33	50.25	15.56	46.58	46.82	14.21	46.16	46.78
	8	10.65	50.29	48.97	10.41	48.37	47.74	10.41	47.69	47.74
tcdb	3	13.42	36.06	36.22	24.48	39.67	38.39	21.90	39.32	37.03
	4	22.74	43.80	41.13	26.83	43.94	42.44	25.81	42.02	41.83
	5	24.99	42.11	41.02	28.08	43.63	43.72	27.57	42.39	42.13
	6	21.94	40.37	41.70	25.05	42.12	44.06	25.03	42.12	43.93
	7	23.06	40.83	42.06	23.05	42.80	43.84	21.70	42.69	43.84
	8	11.81	41.07	40.87	13.61	42.15	41.75	13.61	41.98	41.75

answers to melodic queries. Analysis of the results of individual queries and of retrieval precision (reported elsewhere [12]) shows that for a typical query these similarity measures rank 3 to 4 correct matches in the top 10 results. Putting this result in context, the performance is similar to that of an effective text search engine on typical web data.

When manual queries are used, we have found that separately retaining the melody from each channel, which we call the *all-channels* technique, is far better than any other approach. Overall, both local alignment and n-grams are successful matching techniques, and our results show that music information retrieval is practical and effective.

Acknowledgements

We are grateful to John Harnett, Jodie Lockyer, and Abhijit Chattaraj for their help with the experiments. This work was supported by the Australia Research Council.

References

- [1] I. V. Bakhmutova, V. D. Gusev, and T. N. Titkova. The search for adaptations in song melodies. *Computer Music Journal*, 21(1):58–67, 1997.
- [2] S. Blackburn and D. De Roure. A tool for content-based navigation of music. In *Proc. ACM International Multimedia Conference*. ACM, September 1998.
- [3] J. C. C. Chen and A. L. P. Chen. Query by rhythm an approach for song retrieval in music databases. In *Proceedings of IEEE International Workshop on Research issues in Data Engineering*, pages 139–146. IEEE, 1998.
- [4] D. Deutsch. Grouping mechanisms in music. In D. Deutsch, editor, *The Psychology of Music*, chapter 4, pages 99–134. Academic Press, Inc., 1982.
- [5] J. S. Downie. *Evaluating a Simple Approach to Musical Information Retrieval: Conceiving Melodic N-grams as Text*. PhD thesis, University of Western Ontario, 1999.
- [6] R. Francès. *La Perception de la Musique*. L. Erlbaum, Hillsdale, New Jersey, 1958. Translated by W. J. Dowling (1988).
- [7] T. Kageyama, K. Mochizuki, and Y. Takashima. Melody retrieval with humming. In *Proc. International Computer Music Conference*, 1993.
- [8] K. Lemström and P. Laine. Musical information retrieval using musical parameters. In *Proc. International Computer Music Conference*, pages 341–348, Ann Arbor, 1998.
- [9] R. J. McNab, L. A. Smith, D. Bainbridge, and I. H. Witten. The New Zealand Digital Library MELody inDEX. *Digital Libraries Magazine*, May 1997.
- [10] D. Ó’Maídin. A geometrical algorithm for melodic difference. *Computing in Musicology*, 11:65–72, 1998.
- [11] J. Pickens. A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval. In D. Byrd, J. S. Downie, T. Crawford, W. B. Croft, and C. Nevill-Manning, editors, *International Symposium on Music Information Retrieval*, volume 1, Plymouth, Massachusetts, October 2000.

Table 4: *Eleven-point precision averages for different n-gram lengths using the 28 automatic melody queries and the “count distinct” measure. All channels database.*

		0			9			1			
		008	020	030	008	020	030	008	020	030	
amdb	3	18.43	36.87	36.26	31.07	42.60	41.54	31.15	42.55	42.09	
	4	29.27	44.75	43.74	35.20	46.91	45.93	35.20	46.51	45.20	
	5	29.81	48.10	46.71	35.04	48.93	48.25	35.04	48.93	48.25	
	6	27.32	47.43	46.91	28.86	47.93	47.87	28.86	47.93	47.87	
	7	23.94	43.45	43.65	24.35	44.61	45.77	24.35	44.61	45.77	
	8	12.47	42.61	42.86	12.65	44.31	43.91	12.65	44.31	43.91	
	ecdb	3	13.64	34.56	35.80	18.66	36.96	36.02	18.97	36.63	35.54
		4	23.91	38.29	39.43	27.12	39.31	39.95	26.53	38.24	39.64
5		24.51	38.59	38.34	26.24	38.35	38.15	26.24	38.36	38.15	
6		17.89	39.61	38.35	18.73	38.87	37.71	18.73	38.87	37.71	
7		09.57	37.30	38.81	09.50	37.46	38.20	09.50	37.46	37.73	
8		05.41	37.19	37.59	05.41	37.42	36.31	05.41	37.42	36.31	

- [12] A. L. Uitdenbogerd. *Music Information Retrieval Technology*. PhD thesis, School of Computer Science and Information Technology, RMIT University, 2002.
- [13] A. L. Uitdenbogerd, A. Chattaraj, and J. Zobel. Music information retrieval: Past, present and future. In D. Byrd, J. S. Downie, and T. Crawford, editors, *Current Research in Music Information Retrieval: Searching Audio, MIDI, and Notation*. Kluwer. To appear.
- [14] A. L. Uitdenbogerd and J. Zobel. An architecture for effective music information retrieval. *J. American Society of Information Science and Technology (JASIST)*. To appear.
- [15] A. L. Uitdenbogerd and J. Zobel. Manipulation of music for melody matching. In B. Smith and W. Effelsberg, editors, *Proc. ACM International Multimedia Conference*, pages 235–240, Bristol, UK, September 1998. ACM, ACM Press.
- [16] A. L. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In D. Bulterman, K. Jeffay, and H. J. Zhang, editors, *Proc. ACM International Multimedia Conference*, pages 57–66, Orlando Florida, USA, November 1999. ACM, ACM Press.
- [17] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, California, second edition, 1999.

Table 5: *Eleven point precision averages for local alignment and TF-IDF for n-gram melody matching with n=5, using the set of 28 automatically extracted melody queries. Same extraction method as database.*

		L			a			i		
		008	020	030	008	020	030	008	020	030
amdb	0	31.53	40.21	39.16	32.83	51.86	48.92	35.28	43.94	47.04
	1	31.71	42.94	43.50	00.19	00.41	00.79	33.57	45.92	47.90
	9	31.42	41.74	38.93	34.25	51.93	50.94	35.66	43.66	43.93
	L	31.42	41.68	38.91	27.12	50.64	50.62	35.66	43.67	44.40
ecdb	0	22.07	30.43	26.07	23.74	38.61	39.05	22.27	38.61	37.91
	1	24.64	29.87	28.15	00.41	02.22	03.44	24.92	38.26	37.68
	9	22.05	29.95	27.38	28.38	38.07	37.30	23.56	37.55	39.65
	L	22.05	30.18	28.52	19.21	36.32	37.02	23.65	37.87	39.74
tcdb	0	24.41	32.11	32.56	24.95	35.48	37.36	26.45	35.45	36.38
	1	23.28	27.85	26.38	00.16	00.40	00.66	25.95	31.28	31.45
	9	24.69	32.72	32.68	24.25	35.97	37.57	26.38	34.80	35.64
	L	24.49	32.86	33.06	16.64	31.96	36.37	26.40	34.57	35.50

Table 6: *Eleven point precision averages for local alignment and two variants of TF-IDF for n-gram melody matching with n=5, using the set of 28 automatically extracted melody queries. All channels database.*

		L			a			i		
		008	020	030	008	020	030	008	020	030
amdb	0	24.06	32.05	28.39	25.17	42.11	44.07	24.40	32.11	36.15
	1	20.69	25.91	24.89	00.05	00.10	00.12	24.88	30.75	34.49
	9	24.14	32.87	29.53	24.80	40.42	46.17	24.98	32.92	34.19
	L	24.01	31.72	28.93	04.95	32.39	37.84	25.24	33.09	34.15
ecdb	0	21.53	30.68	26.16	28.14	49.29	47.74	23.84	41.73	43.43
	1	20.00	24.01	22.47	00.03	00.16	00.13	21.02	36.47	39.05
	9	21.59	31.28	26.83	24.40	45.87	46.44	24.36	42.21	42.55
	L	21.63	32.08	27.11	06.46	37.45	42.54	24.57	42.08	42.83
tcdb	0	22.38	32.79	30.73	25.37	43.09	44.18	25.62	40.31	40.16
	1	22.31	26.44	25.94	00.03	00.09	00.13	24.47	34.33	34.58
	9	23.85	34.08	34.05	25.40	42.43	46.80	26.36	39.81	38.51
	L	23.66	34.92	33.33	13.59	36.96	40.61	26.75	39.84	38.45

Table 7: *Eleven-point precision averages for a set of 30 manually-produced melody queries. (a) Different n-gram lengths using and the “count distinct” measure. (b) Local alignment.*

		0			9			1		
		008	020	030	008	020	030	008	020	030
acdb	3	03.98	11.66	12.06	07.48	20.51	24.68	07.11	20.46	26.67
	4	06.19	24.92	25.82	08.89	28.67	31.58	08.64	27.15	32.14
	5	06.14	29.25	34.39	09.19	31.15	34.71	09.19	28.47	35.62
	6	04.45	25.82	34.07	04.11	28.36	34.15	04.11	26.37	34.95
	7	03.71	27.50	32.04	03.71	28.12	32.83	03.71	27.11	31.99
	8	00.11	25.69	31.68	00.11	27.25	32.74	00.11	26.21	31.73
amdb	3	02.78	02.65	02.87	05.45	06.09	05.28	05.21	06.15	06.10
	4	04.41	07.04	05.97	07.06	10.27	08.04	06.99	10.34	08.60
	5	06.16	12.39	10.83	08.29	14.11	13.11	08.08	14.05	12.70
	6	03.26	10.75	11.17	04.11	11.76	12.56	04.11	11.66	12.57
	7	03.15	10.79	10.46	04.05	10.90	10.52	04.05	10.90	10.52
	8	00.11	09.96	10.58	00.11	10.24	10.15	00.11	10.24	10.15
ecdb	3	02.41	05.30	05.25	04.44	09.35	08.54	04.42	09.34	08.70
	4	04.65	12.10	11.71	05.93	14.37	13.63	05.91	13.80	13.50
	5	04.41	13.46	16.73	06.48	15.07	17.34	06.48	15.18	17.27
	6	02.66	12.74	15.03	02.95	12.75	15.61	02.95	12.80	15.39
	7	02.69	10.98	13.61	02.62	11.35	13.83	02.62	11.35	13.90
	8	00.11	09.97	13.26	00.11	10.50	13.96	00.11	10.50	13.51
acdb	a	06.58	36.26	37.06	07.98	28.72	34.92	01.19	20.95	28.62
amdb	a	05.34	13.16	11.59	07.20	13.46	14.28	04.74	12.65	12.84
ecdb	a	05.26	15.56	14.33	06.06	14.76	14.65	04.85	12.36	13.42
tcdB	a	04.45	15.64	17.55	06.50	14.03	17.17	01.39	10.31	14.43

Table 8: *Eleven-point precision averages for local alignment using a set of 30 complete manually-produced melody queries. The headings l, 9 and 0 refer to the type of normalisation applied to the similarity scores. Coordinate matching is shown for n-gram lengths from 3 to 8.*

		a	c					
		1.1.2.1	3	4	5	6	7	8
acdb	0	37.35	12.70	32.95	40.05	37.27	34.00	33.89
	9	32.72	28.13	34.85	40.93	37.71	35.73	34.76
	1	28.10	28.33	34.48	37.99	35.86	35.04	34.81
amdb	0	09.30	02.91	07.80	11.96	10.29	08.90	09.06
	9	09.84	04.93	13.57	15.41	11.71	09.15	09.31
	1	08.77	05.83	13.28	15.42	11.61	09.24	09.28
ecdb	0	15.86	05.37	11.86	17.13	15.58	13.98	13.94
	9	14.92	09.33	14.69	18.55	16.08	13.79	13.93
	1	13.61	10.34	15.07	17.90	15.66	14.00	13.96