A Comparative and Fault-tolerance Study of the Use of N-grams with Polyphonic Music

Shyamala Doraisamy
Dept. of Computing
Imperial College
London SW7 2BZ
+44-(0)20-75948180
sd3@doc.ic.ac.uk

Stefan Rüger
Dept. of Computing
Imperial College
London SW7 2BZ
+44-(0)20-75948355
srueger@doc.ic.ac.uk

ABSTRACT

In this paper we investigate the retrieval performance of monophonic queries made on a polyphonic music database using the n-gram approach for full-music indexing. The pitch and rhythm dimensions of music are used, and the *musical words* (a term coined by Downie [2]) generated enable text retrieval methods to be used with music retrieval. We outline an experimental framework for a comparative and fault-tolerance study of various n-gramming strategies and encoding precision using six experimental databases. For monophonic queries we focus in particular on query-by-humming (QBH) systems. Error models addressed in several QBH studies are surveyed for the fault-tolerance study. Our experiments show that different n-gramming strategies and encoding precision differ widely in their effectiveness. We present the results of our comparative and fault-tolerance study on a collection of 5380 polyphonic music pieces encoded in the MIDI format.

1. INTRODUCTION

With the advances in computer and network technologies, large collections of digital music documents are being created and stored. Managing these large collections requires effective computer-based music information retrieval (Music IR) systems where documents relevant to a user query can be retrieved quickly.

Music documents are stored digitally in many formats and therefore Music IR system designs frequently need to include sophisticated document and query pre-processing modules. These formats have been generally categorized into (i) highly structured formats where every piece of musical information on a piece of musical score is encoded, (ii) semi-structured formats in which sound event information is encoded and (iii) highly unstructured raw audio that encodes only the sound energy level over time. Most current Music IR systems adopt a particular format and therefore queries and indexing techniques are based upon the dimensions of music information that can be extracted or inferred from that particular encoding method [11]. An example would be to input queries by humming in the audio format to a database indexed on a collection of themes encoded using the Parson's code (a simple encoding that reflects only the directions of melodies; each pair of consecutive notes is coded as “U” for up, “D” for down and “R” for repeat, i.e. when pitches are equal) [8]. With query-by-example (QBE), one can input a recording where energy profiles of the query and of the collection of audio recordings are compared. These are, however, still in early research stages [13]. There are several systems that use text-based input modes, of which many are designed for those musically literate, where inputs are music notation keyed into text-boxes [14] or played on a graphically visualised keyboard. These systems usually do include options for those who are not musically literate such as melodic contour inputs similar to the Parson's code. The user would have the additional task of working out the contour using this input mode.

Of the various technologies of query processing and interface, one that has been gaining popularity is query-by-humming (QBH). This can be said to be appealing to a large number of users, whether musically literate or not. A number of studies on QBH have been performed and systems developed in recent years [4-9]. For many who are shy singing in front of others [8], QBH may still be an appealing choice with options such as private listening booths available in music stores that can simply be extended as private QBH booths as well.

A known problem with IR systems in general is query precision, where documents are not retrieved with rank number one due to queries not being specified precisely or just simply erroneous. With Music IR, the user-friendly, or rather music-friendly, queries that are hummed are highly likely to be incorrect. What is required is that Music IR systems (in this case QBH systems) should work for everybody – perfect singers or not [8]. Fault-tolerant or error-tolerant QBH systems are necessary to provide for the large number of unprofessional singers who wish to use the system.

Most of the QBH studies have been based on monophonic music. The query in this context, which is essentially monophonic (unless a choir squeezes onto a microphone!), is made against a collection of monophonic pieces. The vast majority of data collections available today are in the polyphonic form. With automated melody and theme extraction systems still in research stages [3, 15,16], extracting and indexing themes for the development of monophonic databases that can be queried by humming is an onerous task in itself.

We aim to study the querying of a database of polyphonic pieces with a monophonic music sequence using the n-gram approach to full-music indexing. An experimental design is outlined for a comparative study and fault-tolerance study investigating various n-gramming strategies and encoding precision values of the musical n-grams. A survey of error models from several QBH studies is done for the fault-tolerance study.

The paper is structured as follows: Section 2 presents a number of QBH studies and the error models that have been addressed. Section 3 describes the n-gram approach to polyphonic music retrieval and the fault-tolerance of this approach. Section 4 outlines the experimental setup using 5380 polyphonic music pieces and discusses the error model adopted. The retrieval performance evaluation using the precision-at-15 measure is presented in Section 5.

2. QBH AND ERROR MODELS

Work on Music IR systems has, in general, focused more on development rather than evaluation [2]. However, music IR evaluation test-beds and performance measures similar to the Cranfield model in text retrieval are in their early development stages [17]. With a high probability of query imprecision or inaccuracies with QBH, predefined queries, responses and metrics for evaluation would need to be based on QBH error models - how well a system performs under such erroneous inputs. A number of QBH studies have addressed error models [5-7].

In an experimental study by McNab et al on the development of a QBH system [5], ten songs and ten singers were used to get an idea of the kind of input they could expect in their music retrieval system - to find out how people sang well known tunes. The findings had also been used by Downie [2] for the performance evaluation of the n-gram approach towards monophonic music retrieval. The types of errors had been grouped into 4 classes:

1. Expansion – a tendency to expand smaller intervals that fell within 1 and 4 semitones.
2. Compression – a tendency to compress larger intervals that were larger than 5 semitones.
3. Repetition – a tendency to incorrectly repeat notes.
4. Omission – a tendency to simply omit an interval.

The audio query transcription algorithm developed by Haus and Pollastri [6] was based on an error model that is contrary to the idea from the study by McNab et al that singers tend to compress wide leaps and expand sequences of smaller intervals. They assumed constant sized errors based on the idea that every singer has his/her own reference tone in mind and dealt with obtaining this reference tone in their study. Singers would simply sing each note relative to the scale constructed on their own reference tone, apart from some small increases with the size of the interval.

A tempo analysis was done in the QBH study by Kosugi et al [7]. It was observed that singers decided on what tempo to maintain, which was not necessarily the same as that of the original song. The assumption adopted on tempo in their study was that for faster songs there was a tendency for users to choose a tempo that was half the correct one. Fault-tolerance was addressed in the database by making two copies of songs of fast tempos, one at the original tempo and the other at half the tempo.

In these studies, the error models were based on a definition of humming as singing with the syllable *ta*, *da* or *la*, and not whistling or singing with syllables derived from lyrics. Removing lyrics was suggested to remove another possible source of errors that is difficult to quantify [6]. There is also the consideration that, with singing based on lyrics, one could possibly remember the tune better when it is associated with words.

3. N-GRAMS AND FAULT TOLERANCE

N-grams have been widely used in text retrieval, where query terms are decomposed into their constituent n-grams. For example, the word 'music' comprises the bi-grams mu,us,si and ic. A character string formed from n adjacent characters within text is called an n-gram [1]. In music retrieval, an in-depth study of melodic n-grams has been performed by Downie using monophonic music data and interval-only representation [2]. A database of folksongs was converted to an interval-only representation of monophonic *melodic strings*. Using a gliding window, these strings were fragmented into length-n subsections, or windows, called melodic n-grams.

A study on the use of n-grams with polyphonic music retrieval, where the interval-only representation had been extended to include rhythmic information, was done in a previous paper [11]. The strategy is to use all combinations of monophonic musical sequences from polyphonic music data: The gliding window approach is used to divide a music piece into overlapping windows of *n* different adjacent onset times. All possible combinations of melodic strings from each window form musical n-grams. To incorporate interval and rhythm information, for a sequence of *n* onset times, n-grams are constructed in the pattern form of:

[Interval$_1$ Ratio$_1$.. Interval$_{n-2}$ Ratio$_{n-2}$Interval$_{n-1}$]

For a sequence of *n* pitches, an interval sequence is derived with *n-1* intervals by:

$$Interval_i = Pitch_{i+1} - Pitch_i \qquad (1)$$

For a sequence of *n* onset times, a rhythmic ratio sequence is derived with *n-2* ratios obtained by:

$$Ratio_i = \left(\frac{Onset_{i+2} - Onset_{i+1}}{Onset_{i+1} - Onset_i} \right) \qquad (2)$$

Musical words [2] obtained from encoding the n-grams with text letters are used in the indexing and database construction. Queries are processed similarly, which means that the queries might be polyphonic – although we use monophonic queries in this study.

With queries generated from erroneous inputs, such as in QBH, correct retrieval is possible using the n-gram approach to music retrieval. An erroneous query string would generate a number of n-grams that are incorrect out of the total number of n-grams constructed. The probability of retrieving a query correctly would depend on the number of n-grams that are incorrect. Using McNab's error model described in Section 2 and the value of n=4, we illustrate the fault-tolerance of the n-gram approach using the theme from Mozart's Variations in C, K265, "Ah! Vous dirai-je, Maman" (Twinkle Twinkle Little Star), adapted from Barlow and Morgenstern [12] as shown in Figure 1.

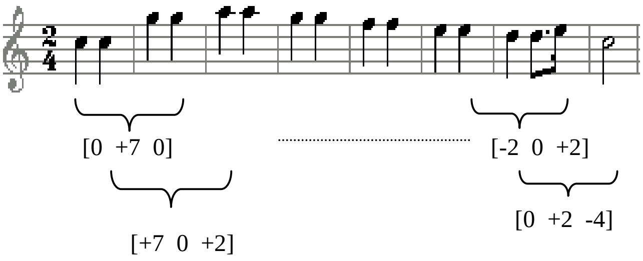

Figure 1. Theme from "Ah! Vous dirai-je, Maman"

In using interval-only representation, n-grams are constructed based on the interval distance (in semitones) and direction using Equation 1. Therefore, the n-gram constructed using the melodic sequence from the first window would be represented as [0 +7 0]. With the gliding window approach, n-grams would be repeatedly generated in this pattern to the end of the excerpt. To obtain 'musical words', the n-grams are encoded using text letters, see for details below.

The set of n-grams generated from Figure 1 are: {[0 +7 0], [+7 0 +2], [0 +2 -2], [+2 0 -2], [0 -2 -2], [-2 0 -2], [0 -2 0], [-2 0 -1], [0 -1 0], [-1 0 -2], [0 -2 0], [-2 0 +2], [0 +2 -4]}.

To illustrate the fault-tolerance, two examples of possible errors, compression and omission, are incorporated into the query string as shown in Figure 2.

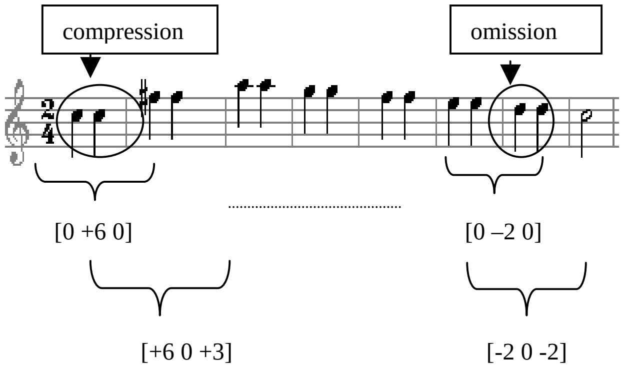

Figure 2. Compression and omission errors

The set of n-grams generated from Figure 2 are: {[0 +6 0], [+6 0 +3], [0 +3 0], [+3 0 -2], [0 -2 0], [-2 0 -2], [0 -2 0], [-2 0 -1], [0 -1 0], [-1 0 -2], 0 -2 0], [-2 0 -2]}.

In this example, around 60% of the n-grams generated from the erroneous input of Figure 2 are similar to those from the excerpt in Figure 1. If this number of n-grams still sufficiently represents the indexed relevant document unambiguously, perfect retrieval is highly feasible.

In our previous study, we had investigated the fault-tolerance of the encoding precision of the n-grams. With large numbers of possible interval values and ratios to be encoded, and a limited number of text representations, classes that clearly represented a particular range of intervals and ratios without ambiguity were investigated. Based on an analysis of the interval distribution, the interval encoding precision was varied based on a mapping function given by :

$$Code = \text{int}\left(X \tanh\left(\frac{Interval_{n-1}}{Y} \right) \right) \qquad (3)$$

In Equation (3), X is a constant set to 27 in this study to limit the code ranges to 26 text letters. With Y = 24, a 1-1 mapping of semitone differences in the range [-13, 13] is obtained. Less frequent semitone differences (which are bigger in size) are squashed and have to share codes. Y determines the rate at which class sizes increase as interval sizes increase. This is a trade-off between classes of small (and frequent) versus large (and rare) intervals. The codes obtained were then mapped to the ASCII character values for letters. To encode the interval direction, positive intervals were encoded as uppercase letters A-Z and negative intervals were encoded with lower case letters a-z and, in the centre, code 0 represented with the numeric character 0.

In this study we continue to adopt the mapping function for the interval encoding in studying fault-tolerance. With the possible errors from the example above, in using a 2-1 mapping, the intervals +6 and +7 would have been encoded with the same text letter. This would have been more fault-tolerant towards the compression error in Figure 2. More detailed interval classification schemes were investigated by Downie [2] and it was concluded in his study that the expected fault tolerance through the application of the classification scheme was not evident. The Parson's code used in the QBH study performed by Prechelt and Typke [8] was said to be highly fault-tolerant on a database of monophonic themes. However, we performed a preliminary test, retrieving from a database of polyphonic pieces using the n-grams approach with Parson's code where hardly any queries were retrieved, hence this approach was not adopted either.

In order to represent the rhythm dimension, we base our ratio encoding on the frequency distribution of ratio values of the data collection. A graph of frequency versus the logarithm of the ratios (onset times were obtained in units of milliseconds) was used to find peaks that clearly identified significant ratios. Ratio bins were constructed based on the midpoints between the identified ratio peaks. These bins provided appropriate quantisation ranges for timing deviations with performance data. Ratio 1 had the highest peak and other peaks occurred in a symmetrical fashion where for every *peak ratio*, there was a symmetrical peak value of 1/*peak ratio*. The peaks identified as ratios greater than 1 were 6/5, 5/4, 4/3, 3/2, 5/3, 2, 5/2, 3, 4 and 5. Ratio 1 was encoded as Z. The bins constructed based on peak ratios greater than 1 were encoded with uppercase letters A-I. Any ratio above 4.5 was encoded as Y. The bins for ratios smaller than 1 based on the peaks 5/6, 4/5, 3/4, 3/2, 3/5, 1/2, 2/5, 1/3, 1/4 and 1/5 were encoded with lowercase letters a-i and y respectively. Wider ratio bins (merging A and B, C and D, etc) were investigated for fault-tolerance with rhythmic deviations.

4. EXPERIMENTAL SETUP

The aim of this study is to test the retrieval performance of querying a polyphonically encoded database with monophonic queries using the n-gram approach to full-music indexing, similar to full-text indexing in text IR. Various n-gramming strategies and encoding precision of the musical n-grams are investigated. A fault-tolerance study based on the error model used in the QBH study of McNab et al [5] is adopted.

A collection of 5380 polyphonic music pieces in the MIDI format was used for the experimental databases. 10 monophonic excerpts extracted from this collection were used as experimental queries. These were popular tunes of various genres, where the main theme of a particular piece was extracted. For the classical pieces, themes were adapted from the Dictionary of Musical Themes [12]. For each query, there were several performances that were considered as documents relevant to the retrieval process in this context. The list of songs and the number of associated relevant documents in the collection are listed in Table 1.

Table 1. Song list

Song ID	**Song Title**	**No. relevant**
1	Alla Turka (Mozart)	5
2	Happy Birthday	4
3	Chariots of Fire	3
4	Etude No. 3 (Chopin)	1
5	Eine kleine Nachtmusik (Mozart)	5
6	Symphony No. 5 in C Minor, (Beethoven)	8
7	The WTC, Fugue 1, Bk 1 (Bach)	2
8	Für Elise (Beethoven)	3
9	Country Gardens	2
10	Hallelujah (Händel)	7

We adopted a similar assumption for relevance as Uitdenbogerd and Zobel [3] where an arrangement was considered to be distinct from other arrangements if one of the following conditions held: (i) it was in a different key, (ii) there was a different number of parts in the arrangement or (iii) there were differences in rhythm, dynamics or structure. Query lengths varied between 15-25 notes for eight of the songs. Song 6 had just 8 notes and song 10 was with 285 notes.

4.1 Database development

Six experimental databases were developed using the value of n=4 with various n-gramming strategies and encoding precision values. The Lemur Toolkit was used in the database development; it is a research-based toolkit that supports the construction of basic text retrieval systems based on language models [10]. Other retrieval models are also supported and these include the vector space model and the probabilistic model. These models use the tf-idf [20] and the Okapi BM25 retrieval function for weighting [19], respectively. Based on several initial tests on the tool using known-item searches, the probabilistic model with the Okapi BM25 function for weighting performed best with musical n-grams and hence was adopted for this study. A description of the databases developed follows:

PR4: The pitch and rhythm dimensions are used for the n-gram construction as described in Section 3. For interval encoding, the value of Y in Equation 3 is set to 24. All bin ranges that had been identified as significant, as listed in Section 3, were used for the ratio encoding.

PR4CA: The pitch and rhythm dimensions are used for the n-gram construction, as described in Section 3. For interval encoding precision that is coarser, the value of Y in Equation 3 is set to 48 for a 2-1 mapping of most intervals smaller than 20 semitones (1 character now covers at least 2 semitones). For coarser ratio encodings, wider ratio bin ranges are used. Half of the text alphabets used with PR4 were used, where one character covers two ratio bins.

AL1: In querying a polyphonic database using a monophonic melodic sequence, n-grams generated from polyphonic documents that are likely to include accompaniment would be matched against n-grams generated from a simple melody. A possible strategy to overcome the problem of n-gramming based on these intercepting accompaniment onsets is to reduce n-grams that are generated from only accompaniment onsets. This database was indexed using n-grams generated from alternate onsets and not every adjacent onset as in the two previous databases. Interval and ratio encoding is similar to PR4.

AL2: Similar to AL1 but n-grams were generated not from every other onset of the gliding window approach but by skipping two onsets.

AM: An approach to automate the development of databases of monophonic melodies from a polyphonic music collection by applying melodic extraction algorithms was proposed by Uitdenbogerd and Zobel [3]. In the study, several melodic extraction algorithms were investigated and the approach in which all top notes were extracted for a given polyphonic music piece performed best. All MIDI channels were combined and the highest note from each simultaneous note event was kept. The method had been referred to as all-mono. We developed this database by n-gramming the highest note from all simultaneous note events and not all possible patterns within a gliding window as described in Section 3. Interval and ratio encoding is similar to PR4.

P4: Only the pitch dimension was used for the n-gram construction. With QBH, there is a high probability of rhythm not being adhered to at all. This database was developed to study the usefulness of retrieving with pitch only. The encoding of pitch is again similar to PR4.

4.2 Error simulation

For the error simulation in our fault-tolerance study, we adopted the error model based on the study by McNab et al [5] as discussed in Section 2. The possible errors on scale differences assumed in the study by Haus and Pollastri [6] and tempo differences by Kosugi et al [7] were not adopted for this study. The use of intervals and rhythmic ratios in the n-gram construction is fault-tolerant to such errors where intervals are invariant to transpositions and scale differences and rhythmic ratios are invariant to augmentation and diminution of tempo.

In the study by Downie [2] that adopted the similar error model based on the study by McNab et al [5], the query lengths and number of notes that were simulated with errors were constant. However, with real queries possibly varying in length, we do not fix the query length. For this initial study using varying query lengths, error probability at 10% and 20% for each query note was investigated.

5. RESULTS

For the performance evaluation, we used the precision-at-15 measure, in which the performance of a system is measured by the number of relevant melodies amongst the first k retrieved, with k=15 in our case. The results are shown as the percentage retrieved from the number of relevant documents for each song query in Table 2. The retrieval performances of the 10 queries are averaged, weighted by the relevant documents for each of the database indexing method in the last column (W.A.). The values have been rounded to the nearest integer.

Table 2. Percentage retrieved with perfect queries

Song ID	PR4	PR4CA	AL1	AL2	AM	P4
1	100	0	0	0	100	0
2	50	25	50	25	50	25
3	0	0	0	0	0	0
4	0	0	0	0	100	0
5	100	40	100	100	100	0
6	13	0	0	0	100	0
7	100	50	100	50	100	0
8	33	33	33	0	67	67
9	50	50	50	50	50	0
10	86	14	71	43	86	0
W.A.	58	18	40	34	80	8

In investigating methods of querying a polyphonic database with a monophonic query, it is clear from the results that preprocessing a polyphonic database for indexing with a melody extraction algorithm is a feasible approach as had been studied previously [3]. On average, 80% of the relevant documents were retrieved within rank 15.

However, looking at full-music indexing of polyphonic music, PR4 performed well despite the large number of index terms generated from n-gramming all possible patterns of polyphonic music data. 58% of the relevant documents were retrieved on average within rank 15. The performance of PR4 clearly indicates that using n-grams in querying a polyphonically encoded database with monophonic query is a promising approach.

In comparing the retrieval performance of the various songs between AM and PR4, two of the largest retrieval measure differences were of Songs 4 and 6. Song 4 had a large number of accompaniment notes interleaved between the melody lines in comparison to the other songs retrieved perfectly by AM, namely Songs 1, 5 and 7. This is one possible reason for the poor retrieval of this song that requires further investigation. The query length of Song 6 of just 8 notes was just not sufficient for retrieval of this large movement of a symphony.

The problem of intercepting accompaniment onsets was not overcome by alternating n-grams with AL1 and AL2. Skipping more onsets would have to be investigated. Other songs not retrieved based on this problem as well were Song 3 and versions of songs 2, 8, 9 and 10. With AL2, short query length as with Song 6 posed a problem where no query document could possibly be generated. The addition of the rhythm dimension clearly improves retrieval, as can be seen from the weak retrieval performance of P4 in comparison to PR4.

Fault-tolerance investigation was performed by simulating errors in the queries with the probability of error levels at 10% and 20% for each of the query notes. The erroneous notes were simulated with errors based on the study by McNab et al [5] using a probability of 0.4 for Repetition, 0.4 for Modification (Expansion or Compression) and 0.2 for Omission. The retrieval performance measures were obtained by averaging the performance results of ten retrieval runs for each song. The results are shown in Table 3 and Table 4.

The retrieval performance for all databases deteriorated under error condition as expected with the increase of erroneous notes. It is also clear from the results that retrieval was not completely lost due to erroneous query notes with the n-gram approach as discussed in Section 3. The performance of AL1 remained almost similar at 40% and 39% under perfect and error conditions (of error probability of 10%) respectively. This fault-tolerance would be further investigated.

Table 3. Percentage retrieved with error probability of 10%

Song ID	PR4	PR4CA	AL1	AL2	AM	P4
1	10	0	0	4	60	0
2	40	5	43	16	50	10
3	0	0	0	0	0	0
4	0	0	0	20	100	0
5	92	28	90	86	92	0
6	10	0	10	0	89	0
7	85	45	85	45	90	0
8	30	30	26	0	47	27
9	50	40	50	40	50	0
10	86	19	71	40	86	0
W.A.	43	14	39	25	70	3

Table 4. Percentage retrieved with error probability of 20%

Song ID	PR4	PR4CA	AL1	AL2	AM	P4
1	0	0	0	0	2	0
2	28	3	23	5	50	0
3	0	0	0	0	0	0
4	0	0	0	0	90	0
5	82	24	84	9	90	0
6	9	0	0	0	78	0
7	70	15	75	35	70	0
8	30	23	7	0	40	0
9	50	40	50	50	50	0
10	86	9	71	26	82	0
W.A.	38	9	32	10	58	0

6. FUTURE WORK

Our experiments suggest following future directions: The problem of poor retrieval for documents with large number of intercepting accompaniment onsets compared to the melody line has to be investigated alongside optimal query lengths. The indexing and query processing strategies would be investigated more exhaustively, such as querying a database indexed with the PR4 method with queries processed with the AL1 and AL2 method. The possibility of results list fusion where each of the independent retrieval strategies is merged in some way that would improve retrieval effectiveness [18] would be investigated.

Further study would be required for rhythmic error models. Error models not just on studies done on QBH but aural and perception studies would be incorporated. Coarser encodings for rhythm and interval information would be further investigated individually for fault-tolerance.

A small query sample has been used for this initial investigation. Larger samples from various genres would have to be investigated independently.

7. CONCLUSIONS

This study shows that full-music indexing of a polyphonic music collection using the n-gram approach is promising. The strategy of using all combinations of monophonic musical sequences from polyphonic music data does not altogether overwhelm the indexing and retrieval of polyphonic music. With large collections of polyphonic music available and the complexity of automating melody extraction for the development of monophonic databases, we have shown a method of retrieving from a polyphonically encoded database without the need to preprocess the database with melody extraction algorithms.

With the high probability of queries generated from erroneous inputs with QBH, a framework for a fault-tolerance study based on QBH error models has been outlined. Results show that a fusion of retrieval strategies may be needed in the development of a more fault-tolerant system.

The use of n-grams with full-music indexing in polyphonic music retrieval is a promising approach for both queries in the polyphonic form as shown in our previous study [11] and with monophonic queries in this study.

8. ACKNOWLEDGEMENTS

This work is partially supported by the EPSRC, UK.

9. REFERENCES

[1] H.S. Heaps, *Information Retrieval: Computational and Theoretical Aspects*, Academic Press, 1978.

[2] J. Stephen Downie, *Evaluating A Simple Approach to Music Information Retrieval: Conceiving Melodic N-Grams As Text*, PhD Thesis, University of Western Ontario,1999.

[3] Alexandra Uitdenbogerd and Justin Zobel, *Melodic Matching Techniques for Large Databases*, ACM Multimedia ’99, Orlando, FL, USA.

[4] Asif Ghias, Jonathan Logan, David Chamberlin and Brian C. Smith, *Query By Humming – Musical Information Retrieval in an Audio Database*, ACM Multimedia ’95 – Electronic Proceedings, San Francisco, CA, Nov. 1995.

[5] Rodger J. McNab, Lloyd A. Smith, Ian H. Witten, Clare L. Henderson and Sally Jo Cunningham, *Towards the Digital Music Library: Tune Retrieval from Acoustic Input*, DL ’96, Bethesda MD USA.

[6] Goffredo Haus and Emanuele Pollastri, *An Audio Front End for Query-by-Humming Systems*, 2nd International Symposium on Music Information Retrieval, ISMIR2001, Indiana, USA, Oct 2001, pp 65-72.

[7] Naoko Kosugi, Yuichi Nishihara, Tetsuo Sakata, Masahi Yamamuro and Kazuhiko Kushima, *A Practical Query-By-Humming System for a Large Music Database*, ACM Multimedia 2000, Los Angeles, CA., Nov. 2000.

[8] Lutz Prechelt and Rainer Typke, *An Interface for Melody Input*, ACM Transactions on Computer-Human Interaction, Vol. 8, No.2, June 2001, pp 133-149.

[9] Pierre-Yves Rolland, Gailius Raskinis and Jean-Gabriel Ganascia, *Musical Content-Based Retrieval: An Overview of the Melodiscov Approach and System*, ACM Multimedia ’99, Orlando, FL, USA.

[10] *Lemur toolkit*, http://www-2.cs.cmu.edu/~lemur

[11] Shyamala Doraisamy and Stefan Rüger, *An Approach Towards A Polyphonic Music Retrieval System*, 2nd International Symposium on Music Information Retrieval, ISMIR2001, Indiana, USA, Oct 2001, pp 187-193

[12] Harold Barlow and Sam Morgenstern, *A Dictionary of Musical themes, London: Ernest Benn*, 1949.

[13] Jonathan Foote, *ARTHUR: Retrieving Orchestral Music by Long-Term Structure*, 1st International Symposium on Music Information Retrieval, ISMIR2000, Massachusetts, USA, Oct 2000.

[14] Andreas Kornstadt, *Themefinder: A Web-Based Melodic Search Tool*, Computing in Musicology 11, 1998, MIT Press.

[15] Lloyd Smith and Richard Medina, *Discovering Themes by Exact Pattern Matching*, 2nd International Symposium on Music Information Retrieval, ISMIR2001, Indiana, USA, Oct 2001, pp 31-32.

[16] Colin Meek and William P. Birmingham, *Thematic Extractor*, 2nd International Symposium on Music Information Retrieval, ISMIR2001, Indiana, USA, Oct 2001, pp 119-128.

[17] J. Stephen Downie, *Wither Music Information Retrieval: Ten Suggestions to Strengthen the MIR Research Community*, 2nd International Symposium on Music Information Retrieval, ISMIR2001, Indiana, USA, Oct 2001, pp 219-222.

[18] Alan F. Smeaton and Francis Crimmins, *Using a Data Fusion Agent for Searching the WWW*, WWW6 Conference, Stanford,USA,1997.

[19] S. Walker, S.E. Robertson, M. Boughanem, G.J.F. Jones, K. Spärck Jones, *Okapi at TREC-6: Automatic ad hoc, VLC, routing, filtering and QSDR*, NIST Special Publication 500-240, The Sixth Text Retrieval Conference (TREC-6), 1998.

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