EXPLORING MICROTONAL MATCHING

Iman S. H. Suyoto and Alexandra L. Uitdenbogerd
School of Computer Science and Information Technology
RMIT University

ABSTRACT

Most research into music information retrieval thus far has only examined music from the western tradition. However, music of other origins often conforms to different tuning systems. Therefore there are problems both in representing this music as well as finding matches to queries from these diverse tuning systems. We discuss the issues associated with microtonal music retrieval and present some preliminary results from an experiment in applying scoring matrices to microtonal matching.

1. INTRODUCTION

In the field of music retrieval research there is a temptation to work only with western music, as this is readily available in a variety of electronic formats. Music of other cultures presents many difficulties. Much is not recorded, and is orally transmitted. Recordings are not found in quantity in the local record store. The unusual tunings of music from some cultures means that it cannot be adequately rendered in standard music formats such as MIDI. Despite these difficulties, there are collections of music gathered by ethnomusicologists that may be available with some effort in transcription.

There has been some work on the representation of music with non-western tuning. Most proposals involve mapping existing notes to slightly different tunings (discussed in Section 3). Additionally there has been comparative analysis of collections of folk music from different cultures. For example, Schaffrath analysed a collection of Chinese folk tunes represented using scale numbers. He used a similar representation for music of German origin and compared the properties of each collection. This cross-comparison ignored the disparate tunings of the two types of music, but allowed for statistical comparison of melodies, revealing telling differences between Chinese and German melodic traditions [28].

At this stage it is not clear whether separate techniques are required for matching microtonally, or whether there is a need for such an application. This work assumes that matching with exact tuning is required, and therefore assumes that the queries will be well-formed, possibly by a musicologist, or that an example recorded piece of music would be used as the source and pitches automatically extracted from it.

In this paper we describe some example tuning systems which are not perfectly represented by standard western representations. We discuss various representations available, then show a simple representation of tuning systems that was used in our work. The experiments show the effect of applying various dynamic programming scoring matrices to the problem of matching. The matching techniques generally work well within the experimental framework, but the test collection is very small due to a lack of availability of microtonal music in electronic form.

2. TUNING SYSTEMS

Many traditional tuning systems exist. In particular, Eastern music has many tuning systems. Some instances are Sundanese (West Java, Indonesia) *slendro* (five-tone system), Javanese (Central and East Java, Indonesia) *pelog* (seven-tone system), and the Chinese five-tone system. For illustration, we present Javanese pelog here. The pitches sound close to the Phyrgian mode, which corresponds to E-F-G-A-B-C-D and its transpositions, in the Western twelve-tone system [18].

Not all tuning systems have corresponding pitches in other tuning system(s). For example, in Sundanese slendro padantara tuning, the interval between two pitches is 240 cents [32]. If we try to fit this into equal-tempered twelve-tone system, we get a pitch in Sundanese slendro padantara that lies almost in the middle of two pitches in equal-tempered twelve-tone system (240 between 200 and 300). Although 240 is closer to 200 than to 300, it is not close enough to any. This is because the difference is larger than *just noticeable difference* (see Section 4.1). This phenomenon is described as *microtonality*.

3. COMPUTER REPRESENTATIONS FOR MICROTONAL MUSIC

For a music representation to accommodate alternative tunings and microtonalism, it needs to provide the facility for the pitches of the tuning system to be defined. We assume here that there is consistency in the tuning across a piece of music or at least a section of the music, so that one tuning definition can be applied to the music.

Below we discuss several methods that can be applied to microtonal representation, and we follow this with the simple approach used for our experimental data.

MIDI MIDI (Musical Instrument Digital Interface), in terms of music format, is a representation that supports polyphonic music and multitrack recording and playback. It encodes notes as integers.

The *Standard MIDI File* (SMF) format and *General MIDI* (GM) defines how MIDI data should be stored. It has no support for non-twelve-tone systems. According to Correia and Selfridge-Field, Scholz has proposed an extension to MIDI concerning tuning systems [4]. However, it is not a strict standard. MIDI Manufacturer Association has published *MIDI TUNING Extensions* [22]. This is however still limited to twelve-tone systems. What users can do is alter the tuning of each pitch. For example, we can detune C by -20 cents, C$\sharp$ by 15 cents, D by 50 cents, and so on. With standard MIDI this can be achieved by pitch bend events.

ESAC ESAC has support for non-twelve-tone systems. For example, Schaffrath defined the tuning of heptatonic scales to allow the encoding of music from many cultures [28].

In ESAC, a pitch is represented as a number. This is applicable to scales originating from most cultures. Encoding is invariant with respect to scale or tuning systems. The tonic (first note) is always represented by "`1`". The note encoding is similar to solfege [29]. Note durations are stored as relative durations. Besides pitches and note durations, the title, source, and social function of a tune are also stored [29].

With regard to melody retrieval, matching on an ESAC-encoded tune can involve pitch components only, note duration component only, or a combination of both. ESAC has been used to identify that the first five notes of the clarinet part of Mozart's second trio from Clarinet Quintet resembles two German folksongs, "Hoert Ihr Herrn und lasst euch sagen" and "Trauer, Trauer, über Trauer" [29].

Despite being effective for melody retrieval, this representation has a basic limitation: it has no support for polyphonic music.

Humdrum Huron has created a set of utilities called Humdrum, which is useful for facilitating the posing and answering of music research questions [19]. Humdrum by itself is not useful for representing any music, because it is a syntax [19]. Humdrum is not limited to any particular tuning system. Users may define their own representations. For twelve-tone-system representation, a Humdrum representation called *kern* has been suggested.

One of the strengths of Humdrum is its support for polyphonic music which is one of its design considerations. It can represent "sequential and/or concurrent time-dependent discrete symbolic data" [19]. Notes that sound at the same time are called *concurrent attributes*, while notes that sound sequentially are called *sequential events* [19].

XML XML is a rapidly developing technology. There is an abundance of tools supporting the meta-language, such as authoring tools, development libraries, viewers, and converters. It can also be extended by users to fit their own needs. Data definition is made possible by the use of a Document Type Description (DTD) and an XML Schema [7]. As XML documents can contain semistructured data [7] they can be used to store music. Using the method suggested by Roland [27], notes are described as shown in the example below.

```
<note pitch="C4" dur="1" />
<note pitch="D4" dur="0.5" />
<note pitch="E4" dur="0.5" />
```

Adding microtonal information would be a simple matter of defining further XML tags that contain tuning information for the scale.

XML is effective, but inefficient in its raw state. However, despite its verbose nature, XML generally compresses very well. Its efficiency can be increased further by indexing.

MTRI The Micro-Tonal Representation for Information Retrieval (MTRI) was designed as a music representation method for our microtonal experiments. It consists of the essential elements required for melody, harmony and rhythm representation, but ignores other musical aspects such as loudness. However, the representation is sufficient to capture the recognisable elements of a piece of music. For the sake of brevity we mainly describe the tuning system representation here, as that is the principle concern of this piece of research.

To encode a tune in MTRI, two files are used: MTP (MTRI pitch specification file) and MTS (MTRI score file). An MTP stores information about pitch frequencies, while an MTS stores information about note events.

In an MTP, the parameter N is used to describe the number of notes in the tuning system. As an example, for the equal-tempered twelve-tone system, $N = 12$ (see Figure 1). A pitch is stored in one line. When storing pitch names that have the same frequency, all the pitch names must appear before the frequency.

An MTS begins with the directive `:Use`, which specifies the MTP to be associated with the tune. For modern compositions with tuning systems that change mid-work, it would be simple to extend the representation by allowing the use of the directive wherever a new tuning system is required. The note names are not limited to the letters A to G, so microtonal melodies that divide the octave into more than 12 notes are also able to be stored.

4. RETRIEVAL

An information retrieval system is only useful when it can answer queries effectively. Information retrieval systems

```
N = 12

C  B+ 261.63
C+ D- 277.18
D     293.66
D+ E- 311.13
E  F- 329.63
F  E+ 349.23
F+ G- 369.99
G     392.00
G+ A- 415.30
A     440.00
A+ B- 466.16
B  C- 493.88
```

Figure 1. MTP for equal-tempered twelve-tone system.

that support ranked queries measure how similar a query is to items in the database according to some meaning of relevance [41]. Most music retrieval systems including the one we report here support ranked queries. In our case, the queries are tunes.

Various matching techniques have been developed to anticipate query vagueness. Dynamic programming was examined by Uitdenbogerd and Zobel [35], a technique suggested by Mongeau and Sankoff [23]. The first published use of n-grams for melody matching was by Downie [10]. The concept was further examined by Uitdenbogerd [35], Pickens [25] and Doraisamy and Rüger [9]. The comparison of both approaches has been shown by Uitdenbogerd and Zobel [36], which indicated that n-grams can be used as a fast alternative to dynamic programming approaches to melody matching without significant loss of effectiveness. An alternative approach is the indexing of notes and applying a look-up of each note in multiple musical keys, with the Chinese remainder theorem for transposition-invariant retrieval [3]. Recent work by Birmingham, Meek, O'Malley, Pardo, and Shifrin [1] uses stochastic models.

Dannenberg, Birmingham, Tzanetakis, Meek, Hu, and Pardo [6] also used HMM (Hidden Markov Models) along with dynamic programming in conjunction with directed modulo-12 standardisation [36] and *Inter Onset Interval ratio* values. They also tested melodic contour matching. Effectiveness was reported as MRR (Mean Reciprocal Rank), the percentage of answers ranked as the first answer, in the top two, and in the top three. Closely related to this work are those by Meek and Birmingham [21] and Shifrin and Birmingham [31], both of which use HMM for searching and MRR to report its effectiveness.

Kageyama, Mochizuki, and Takashima [20] used dynamic programming for their query-by-humming retrieval system. Their system also made use of note duration information for melody matching. The query melody and the melodies in the database were transposed for matching. Note duration was used as the weight for matching score. The effectiveness is reported using the number of melodic samples (out of 100) retrieved as the first answer and in the top ten.

To support comparison of different renditions of the same piece of music, melody standardisation is used [35]. Here, a pitch is not represented exactly as it sounds. This is to support approximate matching. This is analogous to a technique in text retrieval systems called case folding (converting all characters to the same case [41]). There are many possble melody standardisations, but we will only cover the one relevant to this experiment *exact microtonal interval*. We also consider incorporating note duration information for matching.

We use approximate string matching [24]. The algorithm to be used is *edit distance*, also known as *Levenshtein distance*. We do not use the simplest form of this algorithm, as described by Crochemore and Rytter [5]. However, we use its variation called *alphabet-weight edit distance* [12]. Matching is done in conjunction with contour, directed modulo, and exact microtonal interval standardisations. This is discussed further in Section 4.4.

4.1. Pitch Standardisation

Uitdenbogerd's doctoral thesis [34] discusses various retrieval standardisations some of which are the basis for our microtone-enabled techniques. Besides contour and exact interval standardisations, the thesis also focuses on *directed modulo-12* for the underlying experiments. The directed modulo-12 standardisation represents each note as a numeric value which is the interval in semitones (scaled to a maximum of one octave) relative to the previous note. The value is expressed as:

$$\rho_{12} \equiv \begin{cases} 0 & ; d = 0 \\ d(1 + ((I-1) \bmod 12)) & ; d \neq 0 \end{cases} \quad (1)$$

where I is the interval between a note and its previous note (absolute value) and d is 1 if the previous note is lower than the current note, -1 if higher, and 0 if otherwise [33, 34]. This is however limited to twelve-tone systems. For non-twelve-tone systems, the formula can be generalised so that a note is expressed as:

$$\rho_{t} \equiv \begin{cases} 0 & ; d = 0 \\ d(1 + ((I-1) \bmod t)) & ; d \neq 0 \end{cases} \quad (2)$$

where t is the number of tones in the tuning system [33]. This is may only work well for equal-tempered tuning systems and a special scoring technique may need to be developed for matching two tunes having different number of tones in their tuning systems.

Exact Microtonal Interval standardisation is an extension of exact interval standardisation as described in Uitdenbogerd and Zobel [36]. In the exact interval standardisation, a note is represented using the number of semitones between itself and its previous note [36]. In contrast, for microtone-enabled matching purposes, we express intervals in *cents*. As an example, "Melbourne Still Shines" (Figure 2) is represented as "`700 400 100 -500 -500 200 300 -200 -100 -200`" (see Table 1). Two notes that differ are perceived as "fairly similar" when the frequency difference is less than just noticeable difference (JNDF) [8, 26]. JNDF is not a linear measure. At 100 Hz, JNDF is 3 Hz, while at 2 000 Hz, JNDF

Table 1. Exact microtonal interval standardisation example.

Transition	ι_c
C4-G4	700
G4-B4	400
B4-C5	100
C5-G4	-500
G4-D4	-500
D4-E4	200
E4-G4	300
G4-F4	-200
F4-E4	-100
E4-D4	-200

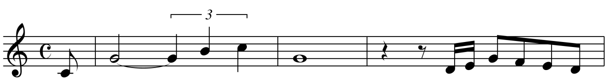

Figure 2. Melbourne Still Shines by ade ishs.

is 10 Hz [26]. However, Zwicker and Fastl [42] suggest using a linear approximation of JNDF function of:

$$\Delta(\nu) = 0.007\nu \tag{3}$$

where ν is frequency in Hz and Δ is the linear JNDF function. Their suggestion is based on musical tones commonly consisting of higher frequency harmonics. In cents, JNDF is approximately 12 cents ($1200 \log_2 |\frac{\nu+0.007\nu}{\nu}| \approx 1200 \log_2 |\frac{\nu-0.007\nu}{\nu}| = 1200(0.01)$).

4.2. Duration Standardisation

Kageyama, Mochizuki, and Takasima [20] suggested the use of dynamic programming for note duration similarity evaluation. They used a weighted sum representing the duration and pitch similarity of melodies. Similarly we use note duration contour standardisation to represent note durations. In this standardisation, a note is represented by its duration relative to its previous note [34]. The following symbols are used: "R" for same, "L" for longer, and "S" for shorter. For example, "Melbourne Still Shines" (see Figure 2) is represented as "`LSRLSRLRRR`".

To incorporate note duration similarity into overall similarity, we model pitch and duration similarities as two orthogonal vectors. Therefore, the overall similarity is:

$$\mathbf{\Sigma} \equiv \varsigma_\pi \hat{\pi} + \varsigma_\delta \hat{\delta} \tag{4}$$

where $\mathbf{\Sigma}$ is the resultant similarity vector, ς_π is the pitch similarity, ς_δ is the duration similarity, and $\hat{\pi}$ and $\hat{\delta}$ are respectively pitch and duration unit vectors. Ranking is based on the magnitude of resultant similarity vector, $|\mathbf{\Sigma}| = \sqrt{\varsigma_\pi^2 + \varsigma_\delta^2}$. If two tunes have the same $|\mathbf{\Sigma}|$, the one with higher ς_π is ranked higher.

4.3. Polyphonic Music

Most music is polyphonic in the sense that more than one note sounds simultaneously. This adds extra complexity to the matching process. In our work we treat each track or part of a polyphonic piece as a separate sequence of notes for matching. For example, if a piece consisted of violin, cello and piano parts, the query would be matched against each of these separately. This results in a similarity score for each part. The best one is chosen as the representative score for the piece. Matching against all tracks in this manner was shown to be an effective approach in earlier work [36]. Where there is polyphony within a part no notes are discarded, and the sequence as defined in the original file is retained. While this may be an issue for matching real queries it does not affect the experiments reported here as they involve known-item searches and the query and potential answers are processed identically.

4.4. Approximate Matching

In this work we use a variation of edit distance called alphabet-weight edit distance. In the "ordinary" edit distance, a penalty score is given for every character difference or *edit operation*. There are three edit operations: *mismatch*, *insertion*, and *deletion* [5, 12, 24]. Each operation has a penalty score of 1. The number of operations must be as minimal as possible. For example, we have two strings: "SHRIMP" and "TRUMPET". The minimal non-match operations are 2 mismatches, 2 insertions, and 1 deletion. Therefore, the distance between "SHRIMP" and "TRUMPET" is $2 + 2 + 1 = 5$.

In *alphabet-weight* edit distance, a scoring matrix is used containing values that should be assigned as costs to various operations [12]. This is commonly applied to genomics, for example, in Henikoff and Henikoff [14]. We tested matching using various scoring matrices. *Local alignment* is a technique to find a substring possessing the highest similarity. This is more useful than *global alignment* (including edit distance), where the overall similarity between two strings is calculated, because it allows a short query to be matched with a long piece of music [35]. Therefore, we use local alignment (also known as Smith–Waterman alignment [12]) for our experiments. In local alignment, there is no negative similarity, and the maximum score is returned as the local alignment score.

For our experiments, we designed several scoring schemes. The scoring schemes are based on the assumption that similar intervals should be penalised less than those that differ greatly. Exact matches should be rewarded highly, and severe mismatches should be penalised highly. For missing notes (insertion/deletion operations), we applied a range of penalties, including zero. We included zero penalty because tunes contain notes that are not significant (such as grace notes), or repetitive notes, of which some can easily be missed or added (for example, "One Note Samba" by Jobim). At the same time, we also consider larger penalties to allow a tighter matching process, which is expected to reduce the num-

Table 2. Scoring schemes for exact microtonal interval.

Name	χ	Insertion/deletion penalty
0_5	0.5	0
0_5-2	0.5	-50
1	1	0
1-2	1	-50
1_5	1.5	0
1_5-2	1.5	-50
2	2	0
2-2	2	-50

		100	100	-180
	0.00	0.00	0.00	0.00
100	0.00	25.00	25.00	1.81
130	0.00	22.51	47.51	22.51
200	0.00	16.72	39.24	41.05
-50	0.00	12.58	14.24	53.47

Figure 3. Similarity between "100 130 200 -50" and "100 100 -180" calculated using exact microtonal interval scoring scheme 1-1.

Table 3. Scoring schemes for note duration contour standardisation.

Name	Insertion/deletion penalty
1	0
2	-1
3	-2

ber of false matches.

For exact microtonal interval standardisation, we make use of the JNDF value in cents as discussed in Section 4.1. We use this formula to calculate reward/penalty scores:

$$\omega = \begin{cases} -T & ; \frac{|\iota_c|}{\Delta_c} \geq 2\left(T^{1/\chi}\right) \\ T - \lfloor\left(\frac{|\iota_c|}{\Delta_c}\right)^{\chi}\rfloor & ; \frac{|\iota_c|}{\Delta_c} < 2\left(T^{1/\chi}\right) \end{cases} \quad (5)$$

where ω is the reward/penalty score, ι_c is the interval in cents, $\chi > 0$ is the reward/penalty order, and $\Delta_c = 1200 \log_2 1.007$. We use $T = 25$ for our experiments. The scoring schemes we use for our experiments are shown in Table 2. As an example, suppose we have two microtonal melodic sequences "100 130 200 -50" and "100 100 -180" and we are to match them with $\chi = 1$ and insertion/deletion score of -25. Figure 3 shows the local alignment matrix for the matching process.

The scoring matrix we used for matching with note duration contour standardisation is given in Figure 4. We used three insertion/deletion penalties (see Table 3).

	L	R	S
L	1	0	-1
R	0	2	0
S	-1	0	1

Figure 4. Scoring matrix for note duration contour standardisation. "L", "R", and "S" respectively indicate a "longer", a "same", and a "shorter".

5. RETRIEVAL PERFORMANCE EVALUATION

We need a measure to evaluate the effectiveness of an information retrieval system. We test the effectiveness of our system using known-item searches. This means, for each query, we already know which specific item we want returned as the answer. A known-item search is similar to the home page finding task in Hawking and Craswell [13], we chose to apply the measures used there, which are *mean reciprocal of rank* (MRR) and the probability that an answer is ranked top 10 (P_{10}).

MRR is also commonly used in question-answering systems (where a correct answer for a question is known), for examples, in Wang, Xu, Yang, Liu, Cheng, Bu, and Bai [40], Voorhees [37], and Voorhees and Tice [39]. Doraisamy and Rüger [9] also used this measure for their music information retrieval experiments. Downie [10] used the term "modified precision" to describe "reciprocal of rank", thus MRR was also used there. MRR has become one of the de facto standard measures for evaluating the performance of music information retrieval systems. This is shown by recent papers by Shifrin and Birmingham [31], Dannenberg, Birmingham, Tzanetakis, Meek, Hu, and Pardo [6], and Meek and Birmingham [21].

MRR can be defined mathematically as $\left\langle \frac{1}{r} \right\rangle$. For example, if three queries produce answers ranked first, fifteenth, and second, the mean reciprocal of rank is $\frac{1}{3}\left(\frac{1}{1} + \frac{1}{15} + \frac{1}{2}\right) = 0.52$. Using the same example, $P_{10} = \frac{1}{3}(1 + 0 + 1) = 67\%$. Higher values of $\left\langle \frac{1}{r} \right\rangle$ and P_{10} indicate more effective retrieval.

MRR and P_{10} might be sufficient to measure the ability of retrieval systems in returning correct answers. However, it is better if a system is able to judge the level of correctness of answers. To test this ability, we use *highest false match* (HFM). HFM is the similarity of the highest-ranked incorrect answer with respect to that of the correct answer [15, 16]. It is typically expressed as percentage. For example, if the similarity score of the correct answer is 40, and the highest ranked incorrect answer is 32, the HFM is $\frac{32}{40} = 80\%$. HFM is useful to determine how well correct answers are separated from incorrect ones. A retrieval system is better than the others when it produces the lowest mean HFM (MHFM) among all. Hoad and Zobel also suggested the use of *separation* and *separation-to-HMF ratio* in their papers [15, 16]. However, we think that it is not necessary for our experiment, since in our experiments, only one answer is considered correct. Those measures would be useful for systems returning more than one relevant answer.

6. EXPERIMENTS

A highly effective technique should have an MRR of 1.00. This means that such a technique always returns a correct answer in the first place. When two techniques each produce an MRR of 1.00 and a P_{10} of 100.00%, we need MHFM to distinguish the two. A technique that produces lower MHFM has a better average ability to separate a correct answer from incorrect ones. Similarly, when a technique does not perform reasonably well, that is, MRR less than 1.00 and P_{10} less than 100.00%, MHFM is also expected to be as low as possible. This means that the first answer returned (which is incorrect) does not mismatch too far.

6.1. Collection and Query Set

Our experiment used as its source a subset of the MIDI file collection used for earlier melody retrieval experiments [36]. As these all use standard western tuning, a set of 3 polyphonic microtonally-tuned pieces were transcribed from recordings, giving a total collection size of 2 390.

From this collection we randomly selected 22 pieces, in addition to the 3 microtonal ones. From each tune, we extracted two random excerpts. These were selected by randomly selecting a track, then randomly choosing a starting point within the track. Polyphonic tracks were handled as described in Section 4.3. Queries were randomly given a length between 12 and 21 notes. Therefore, out of 50 queries, there were 6 microtonal queries.

The query set and the collection contained melodic sequences in equal-tempered twelve-tone, Sundanese degung, and Sundanese madenda tuning systems. Searchable pitch and note duration sequences were derived from MTRI encoding.

6.2. Method

All queries were matched against all of the tunes in our collection. This is regardless of their tuning systems, meaning that microtonal queries were also matched against non-microtonal tunes and vice versa.

In our experiment, we used the scoring schemes described in Section 4.4 for exact microtonal interval pitch standardisations and duration contour standardisation.

Within our experimental framework exact microtonal interval standardisations result into MRR of 1.00 and P_{10} of 100.00%. Therefore, it is worth looking into the values of MHFM. Using exact microtonal interval standardisation always discriminates correct answers from false ones.

Different scoring schemes usually produce different HFM's. We observe that the most contributing factor in lowering the value of HFM is the insertion/deletion penalty score. This can be observed by contrasting:

Table 4. Results of query evaluation for exact microtonal interval standardisation with duration similarity ignored ($\varsigma_\delta = 0$). MHFM and P_{10} are shown as percentage values.

Scoring scheme	MRR	MHFM	P_{10}
`0_5`	1.00	99.77	100.00
`0_5-2`	1.00	91.93	100.00
`1`	1.00	99.51	100.00
`1-2`	1.00	65.75	100.00
`2`	1.00	98.89	100.00
`2-2`	1.00	55.21	100.00

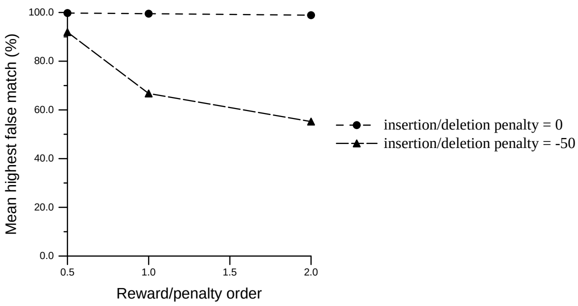

Figure 5. MHFM versus reward/penalty order (χ) with duration similarity ignored ($\varsigma_\delta = 0$).

- exact microtonal interval standardisation scoring schemes `0_5`, and `0_5-2`.
- exact microtonal interval standardisation scoring schemes `1`, and `1-2`.
- exact microtonal interval standardisation scoring schemes `2`, and `2-2`.

There is one exception in our results, however. Using exact microtonal interval standardisation, scoring schemes `0_5-1` and `0_5-2` produce the same result.

For exact microtonal interval standardisation, the value of χ has contribution in discriminating answers. The difference does not seem to be significant in scoring schemes involving zero insertion/deletion penalty. However, combined with large insertion/deletion penalty, the contribution of χ becomes obvious. By comparing scoring schemes `0_5-2`, `1-2`, and `2-2` (all employing insertion/deletion penalty of -50) with the respective MHFM's 91.93%, 65.75%, and 55.21%, we see the effect of increasing χ to reduce MHFM (see Figure 5). We make a further hypothesis that such effect is asymptotic, and it may be investigated in the future.

High measures are shown in the results. This may be caused by the size of our collection, which is small. We may investigate the methods with a bigger collection in the future. The high measures can also be due to known-item search queries in a collection that contains tunes of which melodic patterns are diverse.

Experiments on exact microtonal interval standardisation also confirm the usefulness of duration information

Table 5. Results of query evaluation for exact microtonal interval standardisation with duration similarity incorporated. MHFM is shown as percentage values. P_{10} was always 100% and MRR was always 1.

Scoring scheme	Duration scoring scheme	MHFM
0_5	1	99.66
0_5	2	99.66
0_5	3	99.66
0_5-2	1	91.83
0_5-2	2	91.83
0_5-2	3	91.83
1	1	99.41
1	2	99.41
1	3	99.41
1-2	1	66.71
1-2	2	66.68
1-2	3	66.68
2	1	98.78
2	2	98.78
2	3	98.78
2-2	1	55.26
2-2	2	55.20
2-2	3	55.19

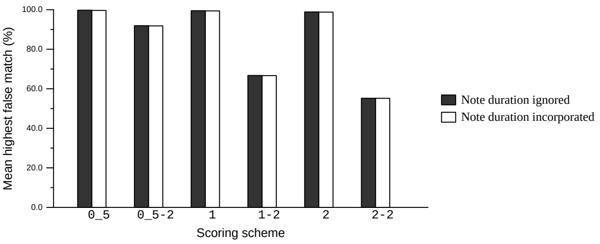

Figure 6. MHFM's for exact microtonal interval standardisation. The bar showing the incorporation of note duration is the best result (lowest MHFM) from three duration contour scoring schemes.

with greater magnitude of insertion/deletion penalty to slightly improve retrieval effectiveness. However, the improvement is insignificant compared to the extra processing required.

7. CONCLUSIONS

Our results demonstrate the applicability of microtone-aware matching techniques to music of various tuning systems. Microtone-aware matching techniques applied in our experiments were non-microtone-aware matching techniques extended for finer frequency spectrum of music.

The results of our experiments show that:

1. Exact microtonal interval standardisation in conjunction with a microtone-aware scoring is effective for microtonal music information retrieval.

2. The most contributing value in lowering mean highest false match of pitch similarity is the insertion/deletion penalty score.

3. In matching with exact microtonal interval standardisation, larger reward/penalty order can cause lower mean highest false match, particularly in conjunction with large insertion/deletion penalty.

4. Note duration information may improve retrieval effectiveness by extending the discrimination between correct/relevant and incorrect/irrelevant answers slightly.

However, we recognise the limitations of the collection and query set used in this experiment. The next step in work on microtonal matching needs to be the procurement of a sufficiently large collection to allow reliable experimentation. Once this is achieved, experiments that demonstrate whether fine-grained techniques are required will be more convincing.

Acknowledgements

We thank Justin Zobel for his feedback and Adinda Marita for help with music transcription.

8. REFERENCES

[1] W. Birmingham, C. Meek, K. O'Malley, B. Pardo, and J. Shifrin. Music information retrieval systems. *Dr. Dobb's Journal*, pages 50–53, September 2003.

[2] 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, USA, October 2000.

[3] M. Clausen, R. Engelbrecht, D. Meyer, and J. Schmitz. PROMS: A web-based tool for searching in polyphonic music. In Byrd et al. [2].

[4] E. Correia, Jr, E. Selfridge-Field, et al. Glossary. In Selfridge-Field [30], pages 581–610.

[5] M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, New York, USA, 1994.

[6] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo. The MUSART testbed for query-by-humming evaluation. In Hoos and Bainbridge [17], pages 41–47.

[7] C. J. Date. *An Introduction to Database Systems*. Addison-Wesley, Boston, USA, eighth edition, 2003.

[8] C. Dodge and T. A. Jerse. *Computer Music: Synthesis, Composition, and Performance*. Wadsworth, Belmont, USA, second edition, 1997.

[9] S. Doraisamy and S. M. Rüger. An approach towards a polyphonic music retrieval system. In Downie and Bainbridge [11].

[10] J. S. Downie. The Musifind music information retrieval project, phase III: Evaluation of indexing options. In *Canadian Assoc. for Inf. Sci. Proc. the 23rd Annual Conf., Connectedness: Information, Systems, People, Organizations*, pages 135–146. CAIS, 1995.

[11] J. S. Downie and D. Bainbridge, editors. *International Symposium on Music Information Retrieval*, volume 2, Bloomington, Indiana, USA, October 2001.

[12] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, UK, 1997.

[13] D. Hawking and N. Craswell. Overview of the TREC-2001 Web Track. In Voorhees and Harman [38], pages 61–68.

[14] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. In *Proc. Natl. Acad. Sci. USA*, volume 89, pages 10915–10919, November 1992.

[15] T. C. Hoad and J. Zobel. Methods for identifying versioned and plagiarized documents. *J. Am. Soc. Inf. Sci. Technol.*, 54(3):203–215, February 2003.

[16] T. C. Hoad and J. Zobel. Video similarity detection for digital rights management. In M. Oudshoorn, editor, *Proc. Australasian Computer Sci. Conf.*, pages 237–245, Adelaide, Australia, February 2003.

[17] H. H. Hoos and D. Bainbridge, editors. *International Symposium on Music Information Retrieval*, volume 4, Baltimore, Maryland, USA, October 2003.

[18] B. Hugh. Claude Debussy and the Javanese Gamelan. `http://cctr.umkc.edu/userx/bhugh/recit/debnotes/gamelan.html`. Accessed 16 December 2003.

[19] D. Huron. Humdrum and kern: Selective feature encoding. In Selfridge-Field [30], pages 375–401.

[20] T. Kageyama, K. Mochizuki, and Y. Takashima. Melody retrieval with humming. In *Proc. Int. Computer Music Conf.*, pages 349–351, 1993.

[21] C. Meek and W. P. Birmingham. The dangers of parsimony in query-by-humming applications. In Hoos and Bainbridge [17], pages 51–56.

[22] MIDI Manufacturer Association. MIDI Tuning Bank and Dump Extensions. `http://www.midi.org/about-midi/tuning_extens.shtml`. Accessed 16 December 2003.

[23] M. Mongeau and D. Sankoff. Comparison of musical sequences. In *Computers and the Humanities*, volume 24, pages 161–175. Kluwer, 1990.

[24] G. Navarro and M. Raffinot. *Flexible Pattern Matching in Strings: Practical On-line Search Algorithms for Texts and Biological Sequences*. Cambridge University Press, Cambridge, UK, 2002.

[25] J. Pickens. A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval. In Byrd et al. [2].

[26] J. G. Roederer. *Introduction to the Physics and Psychophysics of Music*. Springer-Verlag, New York, USA, second edition, 1975.

[27] P. Roland. XML4MIR: Extensible markup language for music information retrieval. In Byrd et al. [2].

[28] 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 95–109. Academic Press, London, UK, 1992.

[29] H. Schaffrath. The Essen Associative Code: A code for folksong analysis. In Selfridge-Field [30], pages 343–361.

[30] E. Selfridge-Field, editor. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press, Cambridge, USA, 1997.

[31] J. Shifrin and W. P. Birmingham. Effectiveness of HMM-based retrieval on large databases. In Hoos and Bainbridge [17], pages 33–39.

[32] A. Supandi, U. Ngalagena, I. Djunaedi, D. Sain, and R. S. Riswara. *Teori Dasar Karawitan*. Pelita Masa, Bandung, Indonesia, third edition, 1976.

[33] I. S. H. Suyoto, S. Uitdenbogerd, and J. Zobel. Microtonal music information retrieval. Research paper submission for RMIT School of Computer Science and Information Technology "Research Methods" course (unpublished), 2003.

[34] A. L. Uitdenbogerd. *Music Information Retrieval Technology*. PhD thesis, School of Computer Science and Information Technology, RMIT, Melbourne, Australia, 2002.

[35] A. L. Uitdenbogerd and J. Zobel. Matching techniques for large music databases. In D. Bulterman, K. Jeffay, and H. J. Zhang, editors, *Proc. ACM Multimedia Conf.*, pages 57–66, Orlando, USA, November 1999.

[36] A. L. Uitdenbogerd and J. Zobel. Music ranking techniques evaluated. In M. Oudshoorn, editor, *Proc. Australasian Computer Sci. Conf.*, pages 275–283, Melbourne, Australia, January 2002.

[37] E. M. Voorhees. Overview of the TREC-9 question answering track. In E. M. Voorhees and D. K. Harman, editors, *Proc. Ninth Text REtrieval Conf.*, pages 71–79, Gaithersburg, USA, November 2000. National Institute of Standards and Technology.

[38] E. M. Voorhees and D. K. Harman, editors. *Proc. Tenth Text REtrieval Conf.*, Gaithersburg, USA, November 2001. National Institute of Standards and Technology.

[39] E. M. Voorhees and D. M. Tice. The TREC-8 question answering track evaluation. In E. M. Voorhees and D. K. Harman, editors, *Proc. Eighth Text REtrieval Conf.*, pages 83–105, Gaithersburg, USA, November 1999. National Institute of Standards and Technology.

[40] B. Wang, H. Xu, Z. Yang, Y. Liu, X. Cheng, D. Bu, and S. Bai. TREC-10 experiments at CAS-ICT: Filtering, web and QA. In Voorhees and Harman [38], pages 109–121.

[41] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, San Fransisco, USA, second edition, 1999.

[42] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models*. Springer-Verlag, Berlin, Germany, second edition, 1999.