Effectiveness of Note Duration Information for Music Retrieval

Iman S. H. Suyoto and Alexandra L. Uitdenbogerd

School of Computer Science and Information Technology, RMIT
GPO Box 2476V, Melbourne, Victoria 3001, Australia
imsuyoto@cs.rmit.edu.au, sandrau@rmit.edu.au

Abstract. Content-based music information retrieval uses features extracted from music to answer queries. For melodic queries, the two main features are the pitch and duration of notes. The note pitch feature has been well researched whereas duration has not been fully explored. In this paper, we discuss how the note duration feature can be used to alter music retrieval effectiveness. Notes are represented by strings called standardisations. A standardisation is designed for approximate string matching and may not capture melodic information precisely. To represent pitches, we use a string of pitch differences. Our duration standardisation uses a string of five symbols representing the relative durations of adjacent notes. For both features, the Smith-Waterman alignment is used for matching. We demonstrate combining the similarity in both features using a vector model. Results of our experiments in retrieval effectiveness show that note duration similarity by itself is not useful for effective music retrieval. Combining pitch and duration similarity using the vector model does not improve retrieval effectiveness over the use of pitch on its own.

1 Introduction

The field of music information retrieval (MIR) research explores ways in which users can better find pieces of music in which they are interested. For *content-based* MIR, we attempt to find answers to queries that contain a fragment of music. This music fragment can be of two main types: an audio sample or a set of notes. The goal of the user could be to find the exact piece of music that they have heard, or to find music that is similar, such as might occur in copyright infringement or in arrangements of a piece. The latter is our main interest in this research.

Current state of the art in content-based MIR has user queries consisting of sung or symbolically created queries. The ability to extract melodies from an audio stream consisting of a single voice is at an acceptable level of precision for matching. The same cannot be said as yet of note extraction from typical commercial recordings of music. Thus for melody search we mainly work with collections of symbolically represented music, such as found in Musical Instrument Digital Interface (MIDI) files.

A technique that has been shown to work reasonably well [1] is a three-phase matching process. First, as most pieces of music are *polyphonic*, that is, have more than one note sounding at the same time, representative melodies or themes are extracted from each piece in the collection. Second, both the pieces and queries are transformed into a standardised form that retains the salient features for matching and allows straightforward matching. Third, a similarity measure is applied to determine the amount of match for each piece, resulting in a ranked set of answers. Melody matching gives quite good results when a simple string representation of the pitch of extracted melodies is compared. While there has been work previously using both pitch and rhythm (see for example Kageyama, Mochizuki, and Takashima [2], McNab et al. [3], Chen and Chen [4], Lemström, Laine, and Perttu [5], and Dannenberg et al. [6]), the relative value of these two aspects of melody for matching have not been quantified for polyphonic collections, and whether a string-matching approach is of benefit in this situation. The experiments reported in this paper show that rhythm, when expressed using an alphabet of five relative values, is quite poor in its own right for matching, even more so than a three-value alphabet representation of a melody's pitch contour. Further, when combined using a vector model, it does not improve the precision of retrieved answers to queries.

Below we discuss the different melody standardisations used in our experiments (Sec. 2), the dynamic-programming-based matching technique we applied (Sec. 3), and the experiments that show that simple pitch matching is superior to a vector-combined pitch and rhythm approach (Sec. 4).

2 Standardisations

To support approximate matching, we convert the melody into searchable representations called *standardisations*. A standardisation is designed for approximate string matching and may not capture melodic information precisely [1]. In this paper, we discuss three pitch standardisations and one duration standardisation. The three pitch standardisations are *contour*, *extended contour*, and *directed modulo-12* (see Secs. 2.1, 2.2, and 2.3). For duration, we use both the *contour* and *extended contour* standardisation (see Sec. 2.4).

2.1 Pitch contour standardisation

The pitch contour standardisation uses three distinct symbols to represent a note. The symbols represent the movement direction of the previous note pitch to the current note pitch [7]. We use the convention "`S`" for same, "`U`" for up, and "`D`" for down. The first note is not represented. For example, the melody shown in Fig. 1 is represented as "`UUUDDUUDDD`".

2.2 Pitch extended contour standardisation

For finer granularity, the pitch contour standardisation is extended so that there are *small* and *big* up's (symbolised as "`u`" and "`U`", respectively) and down's

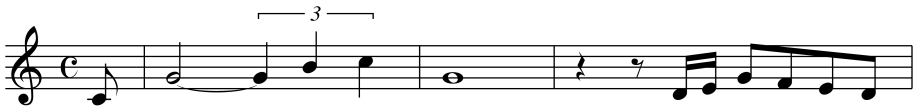

Fig. 1. "Melbourne Still Shines" by ade ishs.

("`d`" and "`D`"). We use three or more semitones as big intervals. For example, the melody shown in Fig. 1 is represented as "`UUuDDuUddd`".

2.3 Pitch directed modulo-12 standardisation

The directed modulo-12 standardisation uses direction information too. A note is represented as a value ρ_{12} which is the interval between a note and its previous note scaled to a maximum of one octave [7, 8]:

$$\rho_{12} \equiv d(1 + ((I - 1) \bmod 12)) \tag{1}$$

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. For example, the melody shown in Fig. 1 is represented as "`7 4 1 -5 -5 2 3 -2 -1 -2`"[1].

2.4 Duration contour and extended contour standardisations

Just as in pitch contour-based standardisations, the duration contour and extended contour standardisations also employ three and five distinct symbols respectively to represent a note. In the case of duration, we use "`S`", "`s`", "`R`", "`l`", and "`L`" for "much shorter", "a little shorter", "same", "a little longer", and "much longer" respectively . (Analogous to pitch contour standardisation, the duration contour standardisation does not have "`s`" and "`l`" symbols). The quantisation we use is based on the encoding in Moles [9]. Let λ_C be the current note, λ_P be the previous one, and $K = \lambda_C/\lambda_P$. A note is represented based on the ranges of $\log_2 K$ as illustrated in Fig. 2. For example, the melody shown in Fig. 1 is represented as "`LSRLSRlRRR`".

3 Retrieval

The use of duration information along with dynamic programming was suggested by Kageyama, Mochizuki, and Takasima [2]. They suggest that note durations be used as penalty scores for insertion and deletion operations. How the scores are calculated is however not formally defined. In this work, we also use a dynamic programming approach. In particular, we use the Smith–Waterman alignment [10] (also known as *local alignment* [11]) which is useful to find a substring

[1] Note that a figure is treated as a symbol. Therefore, it is a 10-symbol string.

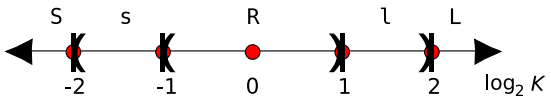

Fig. 2. Duration extended contour quantisation. $K = \lambda_C/\lambda_P$ where λ_C and λ_P are respectively the current and previous note durations.

		S	S	D	U	U
	0	0	0	0	0	0
U	0	0	0	0	2	2
U	0	0	0	0	1	4
D	0	0	0	2	1	3
S	0	2	2	1	1	2

Fig. 3. Local alignment between "UUDS" and "SSDUU".

with highest similarity. Because query tunes typically translate to short strings while tunes in the collection typically to long strings, the alignment is more suitable than global alignment [1].

To calculate the local alignment between two strings s and q, we perform the following steps:

1. Prepare the data structure.
   (a) Construct a matrix of which dimension is $(|s|+1)\times(|q|+1)$. We use 0 as the base index, i.e. the column indices are $0, 1, 2, \ldots, |q|$, the row indices are $0, 1, 2, \ldots, |s|$, and the symbol indices for s and q are respectively $0, 1, 2, \ldots, |s|-1$ and $0, 1, 2, \ldots, |q|-1$.
   (b) Initialise the 0-th row and column with 0.
2. Calculate the score.
   (a) For i in $\langle 1 \ldots |s| \rangle$:
      i. For j in $\langle 1 \ldots |q| \rangle$:
         A. $D_{i,j} \leftarrow \max(0, D_{i-1,j} + I, D_{i,j-1} + I, D_{i-1,j-1} + M(s_{i-1}, q_{j-1}))$ where I is the insertion/deletion score (commonly non-positive) and M is the match/mismatch function. The values for M and I that we use in our experiments are detailed in Sec. 4.

The local alignment score is $\max(D_{i,j}); i \in \{1 \ldots |s|\}, j \in \{1 \ldots |q|\}$. For example, suppose $s =$ UUDS, $q =$ SSDUU, $M(x,x) = 2$ (a match), $M(x,y)|_{x \neq y} = -2$ (a mismatch), and $I = -1$. The matrix looks like the one shown in Fig. 3. The local alignment score is the maximum score in the matrix, i.e. 4.

We are experimenting with a vector model to combine similarity evidences from both pitch and duration matching. The pitches and durations are symbolised by the respective standardisations. As vectors, they are modelled as being perpendicular to each other. The overall similarity is indicated by the resultant

similarity vector. The following formula is based on one in our previous work [12], except that now we also assign weights for both pitch and duration components:

$$\mathbf{\Sigma} \equiv w_\pi \varsigma_\pi \hat{\pi} + w_\delta \varsigma_\delta \hat{\delta} \tag{2}$$

where $\mathbf{\Sigma}$ is the resultant similarity vector, ς_π is the pitch similarity, ς_δ is the duration similarity, w_π and w_δ are both weight constants, and $\hat{\pi}$ and $\hat{\delta}$ are respectively pitch and duration unit vectors. Ranking is then based on the magnitude of resultant similarity vector, $|\mathbf{\Sigma}| = \sqrt{w_\pi^2 \varsigma_\pi^2 + w_\delta^2 \varsigma_\delta^2}$. Therefore, the value of w_π is not meaningful on its own, and neither is w_δ. However, the ratio w_π / w_δ (or reciprocally, w_δ / w_π) is.

4 Experiments

Our aim with these experiments was to determine how effective rhythm information is for melody retrieval using our experimental framework of a polyphonic MIDI file collection, manual queries, and two sets of relevance judgements [13].

The collection consists of 14,193 MIDI files, which are a superset of those used in our earlier experiments (such as Uitdenbogerd and Zobel [1, 14], and Uitdenbogerd, Chattaraj, and Zobel [13]). The query set used here is the set of 28 manual melody queries created by a musician upon listening to a set of rendered polyphonic pieces. We used two sets of relevance judgements. The first, known as *automatic*, was created by Uitdenbogerd by identifying likely matches by file-name, and verifying by listening. The second, called *manual*, was the result of pooling top answers from several matching techniques, and asking users to decide upon listening whether the pieces were similar. More detail is found in Uitdenbogerd, Chattaraj, and Zobel [13].

As a baseline of our experiment, for pitch matching, we use $M(x, x) = 1$ for a match, $M(x, y)|_{x \neq y} = -1$ for a mismatch, and $I = -2$ for an insertion/deletion (see Sec. 3) as used in Uitdenbogerd and Zobel [1]. For duration matching, we use 39 scoring matrices. The scoring matrices are obtained by varying the variables $a, b, c, \ldots, i$ shown in Fig. 4 as detailed in Table 1. The matrix means if there is a match "`S`"-"`S`", $M(\mathtt{S}, \mathtt{S}) = c$; a mismatch "`S`"-"`s`", $M(\mathtt{S}, \mathtt{s}) = d$; etc. At any time, $a \geq b \geq c \geq d \geq e \geq f \geq g \geq i \geq h$.

5 Retrieval Performance Evaluation

The queries in our experiments are topic-oriented, i.e. for one query there can be more than one relevant answer.

To evaluate the effectiveness of every matching method, we use a standard measurement technique for such a task, i.e. using *precision* and *recall*:

$$P \equiv \frac{|\mathbf{Rel} \cap \mathbf{Ret}|}{|\mathbf{Ret}|} \tag{3}$$

$$R \equiv \frac{|\mathbf{Rel} \cap \mathbf{Ret}|}{|\mathbf{Rel}|} \tag{4}$$

	S	s	R	l	L
S	c	d	f	i	h
s	d	b	e	g	i
R	f	e	a	e	f
l	i	g	e	b	d
L	h	i	f	d	c

Fig. 4. Scoring matrix for duration extended contour standardisation. "S", "s", "R", "l", and "L" respectively indicate a "much shorter", an "a little shorter", a "same", an "a little longer", and a "much longer".

Table 1. Scoring schemes for duration extended contour standardisation. For all scoring schemes, $a \geq b \geq c \geq d \geq e \geq f \geq g \geq i \geq h$.

Scoring scheme	a	b	c	d	e	f	g	h	i
1	1	1	1	1	−1	−1	−1	−1	−1
2	2	1	1	1	−1	−1	−1	−1	−1
3	3	1	1	1	−1	−1	−1	−1	−1
4	3	2	1	1	−1	−1	−1	−1	−1
5	3	3	1	1	−1	−1	−1	−1	−1
6	3	3	2	1	−1	−1	−1	−1	−1
7	3	3	3	1	−1	−1	−1	−1	−1
8	3	3	3	3	−1	−1	−1	−1	−1
9	3	3	3	2	−1	−1	−1	−1	−1
10	3	3	3	1	−1	−1	−1	−1	−1
11	3	3	3	0	−1	−1	−1	−1	−1
12	3	3	3	−1	−1	−1	−1	−1	−1
13	3	2	1	0	−2	−3	−3	−3	−3
14	3	2	1	0	−3	−3	−3	−3	−3
15	3	2	1	0	−1	−2	−3	−3	−3
17	3	2	1	0	−1	−1	−3	−3	−3
17	3	2	1	0	−1	−1	−2	−3	−3
18	3	2	1	0	−1	−1	−1	−3	−3
19	3	2	1	0	−1	−1	−1	−3	−2
20	3	2	1	0	−1	−1	−1	−3	−1
21	3	2	1	0	−1	−1	−1	−2	−1
22	3	2	1	0	−1	−1	−1	−1	−1

where P is precision, R is recall, **Rel** is the set of relevant tunes and **Ret** is the set of retrieved tunes. Precision can be averaged at 11 recall levels, $0.0, 0.1, 0.2, \ldots, 1.0$, to obtain the *11-point recall-precision average* [15]:

$$\left\langle \overline{P}(r) \right\rangle_{r=0.0,0.1,0.2,\ldots,1.0} = \frac{\sum_{r=0}^{10} \sum_{i=1}^{N_q} \frac{P_i(0.1r)}{N_q}}{11} \tag{5}$$

which is the measure we use to compare the effectiveness of the techniques in our experiments. However, since some queries have less than 11 relevant answers, we use *interpolated precision* values, which can be calculated using the following formula [15]:

$$P(j) = \max_{j \le r \le j+0.1} P(r) \tag{6}$$

where $j \in \{0.0, 0.1, 0.2, \ldots, 0.9\}$. Higher 11-point recall-precision average means more effective retrieval technique.

6 Results and Analysis

In our experiment, queries were matched against all tunes in our collection 23 times, once for pitch matching using the directed modulo-12 standardisation and 22 times for duration matching using the 22 scoring schemes.

To combine pitch and duration similarities using Eq. 2, We used six different w_π/w_δ values: 0, 1, 3, 5, 7, and ∞. The last one is the baseline performance, i.e. duration information is ignored ($w_\delta = 0$), whereas the first one means pitch information is ignored ($w_\pi = 0$).

For automatic relevance judgments, the baseline performance is an 11-point recall-precision value of 52.15%. The results of using other w_π/w_δ values are shown in Tables 2. For manual relevance judgments, the baseline performance is an 11-point recall-precision value of 51.84%. The results of using other w_π/w_δ values are shown in Tables 3.

Taking the best results from each w_π/w_δ value, we obtain the graph shown in Fig. 5. It shows that the peak performance is obtained when $w_\pi/w_\delta = 5$.

From both relevance judgments, duration information by itself is shown to be not useful for retrieval. In our experiments with automatic relevant judgments, duration information does not improve retrieval performance over that using pitch information per se, whereas with manual relevance judgments using $w_\pi/w_\delta = 5$ and scoring schemes `16`, `17`, `18`, and `19`, slightly better performance is obtained. We analyse further whether duration matching improves retrieval effectiveness using Wilcoxon signed-rank test with one-sided confidence level (α) of 0.05. The null hypothesis is that duration information does not improve retrieval effectiveness; with alternative hyptothesis that duration information does improve retrieval effectivenes. It is found that incorporating duration information using the vector model does *not* imply significant performance gain.

To see how much information is actually contained in the standardised strings of the tunes in our collection, we compress the strings. The rationale behind this

Table 2. 11-point recall-precision percentage values for automatic relevance judgments.

Baseline performance = 52.15.

	w_π/w_δ				
Scoring scheme	0	1	3	5	7
1	0.87	26.40	49.73	51.32	51.32
2	2.81	18.86	47.92	50.89	51.49
3	1.71	12.67	42.41	51.40	51.41
4	2.22	8.93	38.49	51.23	51.06
5	2.57	5.46	36.00	47.63	49.72
6	2.87	4.18	37.45	46.19	49.72
7	2.47	3.59	36.78	48.39	49.41
8	4.08	7.38	37.56	45.51	50.20
9	3.27	4.36	35.52	46.16	49.50
10	2.47	3.59	36.78	48.39	49.41
11	3.84	2.16	33.48	49.50	50.70
12	3.57	2.07	34.98	50.95	50.86
13	1.35	4.74	36.61	50.23	50.14
14	1.84	4.80	36.51	50.53	50.42
15	1.34	4.09	36.58	51.24	51.07
16	1.16	4.66	34.98	51.95	51.27
17	1.16	4.66	34.98	51.95	51.27
18	1.16	4.66	34.98	51.95	51.27
19	1.16	4.66	34.98	51.95	51.27
20	1.16	4.66	34.94	51.93	51.27
21	1.16	4.66	34.94	51.93	51.27
22	1.04	4.55	34.93	51.93	51.27

Table 3. 11-point recall-precision percentage values for manual relevance judgments.

Baseline performance = 51.84.

	w_π/w_δ				
Scoring scheme	0	1	3	5	7
1	0.94	25.24	50.52	52.14	52.14
2	2.60	20.38	48.81	52.04	52.65
3	1.18	13.67	42.87	52.60	52.96
4	1.05	7.83	39.83	52.91	53.13
5	0.67	3.73	36.45	47.23	51.49
6	1.05	3.89	35.95	49.10	51.61
7	0.79	3.84	33.95	48.70	49.81
8	3.57	7.65	36.90	45.49	50.96
9	1.64	4.52	33.71	47.21	50.54
10	0.79	3.84	33.95	48.70	49.81
11	0.40	2.72	31.04	48.30	51.36
12	0.00	2.22	33.62	50.19	52.45
13	0.48	5.40	37.93	52.09	51.54
14	0.49	5.40	37.57	52.03	51.65
15	0.89	4.93	38.30	52.96	53.14
16	0.71	5.71	36.72	53.72	53.42
17	0.71	5.71	36.72	53.72	53.42
18	0.71	5.71	36.72	53.72	53.42
19	0.71	5.71	36.72	53.72	53.42
20	0.71	5.70	36.68	53.70	53.41
21	0.71	5.70	36.68	53.70	53.41
22	0.60	5.57	36.65	53.70	53.41

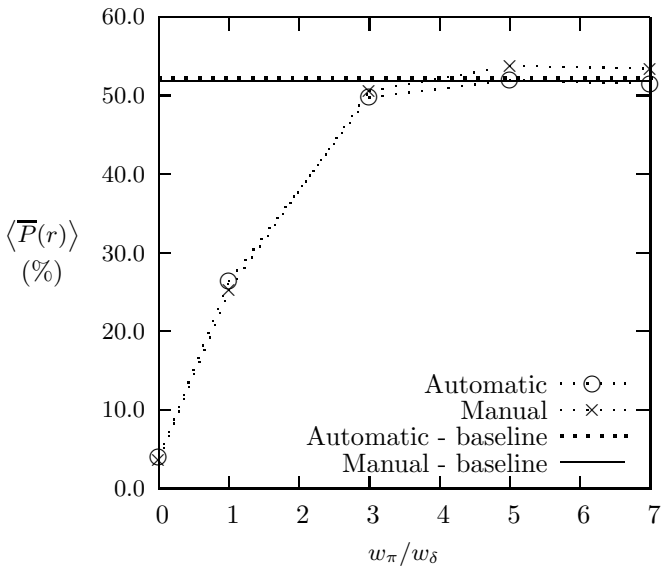

Fig. 5. Best 11-point recall-precision values.

is that strings that contain more information (thus having higher entropy) are less compressible than those containing less information. We compress the strings using the bzip2 program[2]. In uncompressed state, pitch information occupies 35.85 megabytes and so does duration information. The compressed standardised string sizes are shown in Table 4. That duration extended contour strings are more compressible than pitch contour strings reflects that not much information is contained if tunes are represented only by their note durations despite the larger alphabet size.

7 Conclusion and Future Work

This paper inspects the performance of combining pitch and duration similarities using a vector model. The results of our experiment show that:

1. Duration information on its own is not useful for music retrieval.
2. The vector model is not appropriate to combine pitch and duration similarities for the purpose of improving retrieval effectiveness over the use of pitch information on its own.

Rhythm seems to be insufficiently varied for it to be useful for melody retrieval. However, the combination of pitch and rhythm is sometimes needed in order for humans to distinguish or identify melodies. Using a representation that

[2] see `http://sources.redhat.com/bzip2/`

Table 4. Compressed standardised string sizes.

Uncompressed size = 35.85 megabytes.

Standardisation	Compressed size (megabytes)	Compression ratio (%)
Pitch directed modulo-12	8.15	22.74
Pitch extended contour	6.75	18.83
Pitch contour	3.95	11.02
Duration extended contour	3.83	10.69

combines the pitch and rhythm in a manner that preserves the relative position of the match in each case may yield better results. This should be subject to further experimentation.

References

1. Uitdenbogerd, A.L., Zobel, J.: Melodic matching techniques for large music databases. In Bulterman, D., Jeffay, K., Zhang, H.J., eds.: Proc. ACM Multimedia Conf., Orlando, USA (1999) 57–66
2. Kageyama, T., Mochizuki, K., Takashima, Y.: Melody retrieval with humming. In: Proc. Int. Computer Music Conf. (1993) 349–351
3. McNab, R.J., Smith, L.A., Witten, I.H., Henderson, C.L., Cunningham, S.J.: Towards the digital music library: Tune retrieval from acoustic input. In: Proc. ACM Digital Libraries. (1996)
4. Chen, J.C.C., Chen, A.L.P.: Query by rhythm: An approach for song retrieval in music databases. In: Proc. IEEE Int. Workshop on Research Issues in Data Engineering. (1998) 139–146
5. Lemström, K., Laine, P., Perttu, S.: Using relative interval slope in music information retrieval. In: Proc. Int. Computer Music Conf., Beijing, China (1999) 317–320
6. Dannenberg, R.B., Birmingham, W.P., Tzanetakis, G., Meek, C., Hu, N., Pardo, B.: The MUSART testbed for query-by-humming evaluation. In Hoos, H.H., Bainbridge, D., eds.: Proc. Inf. Conf. Music Inf. Retrieval, Baltimore, USA (2003) 41–47
7. Uitdenbogerd, A.L.: Music Information Retrieval Technology. PhD thesis, School of Computer Science and Information Technology, RMIT, Melbourne, Australia (2002)
8. Suyoto, I.S.H.: Microtonal music information retrieval. Master's thesis, School of Computer Science and Information Technology, RMIT, Melbourne, Australia (2003)
9. Moles, A.: Information Theory and Esthetic Perception. University of Illinois Press, Urbana, US (1966)
10. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. **147** (1981) 195–197
11. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge, UK (1997)

12. Suyoto, I.S.H., Uitdenbogerd, A.L.: Exploring microtonal matching. In Buyoli, C.L., Loureiro, R., eds.: Proc. Inf. Conf. Music Inf. Retrieval, Barcelona, Spain (2004) 224–231
13. Uitdenbogerd, A.L., Chattaraj, A., Zobel, J.: Methodologies for evaluation of music retrieval systems. (INFORMS J. Computing) Originally presented at ISMIR 2000; to appear.
14. Uitdenbogerd, A.L., Zobel, J.: Music ranking techniques evaluated. In: Proc. Australasian Computer Sci. Conf., Melbourne, Australia (2002) 275–283
15. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, New York, USA (1999)