TIME SERIES ALIGNMENT FOR MUSIC INFORMATION RETRIEVAL

Norman H. Adams, Mark A. Bartsch, Jonah B. Shifrin, Gregory H. Wakefield
EECS Dept., University of Michigan
Ann Arbor, MI 48109-2110
nhadams@umich.edu

ABSTRACT

Time series representations are common in MIR applications such as query-by-humming, where a sung query might be represented by a series of 'notes' for database retrieval. While such a transcription into a sequence of (pitch, duration) pairs is convenient and musically intuitive, there is no evidence that it is an optimal representation. The present work explores three time series representations for sung queries: a sequence of notes, a 'smooth' pitch contour, and a novel sequence of pitch histograms. Dynamic alignment procedures are described for the three representations. Multiple continuity constraints are explored and a modified dynamic alignment procedure is described for the histogram representation. We measure the performance of the three representations using a collection of naturally sung queries applied to a target database of varying size. The results show that the note representation lends itself to rapid retrieval whereas the contour representation lends itself to robust performance. The histogram representation yields performance nearly as robust as the contour representation, but with computational complexity similar to the note representation.

1. INTRODUCTION

Time series representations are ubiquitous in information retrieval applications [14, 27]. MIR is no exception; sequences of notes, pitch estimates, or MFCCs, for example, are common [5, 6, 16]. It is well established that for comparing two time series, a direct comparison, such as the Euclidean distance, yields a brittle metric [13, 27]. A similarity metric that is robust to elastic shifts and scales of the time index is required. This has reinvigorated interest in dynamic time warping (DTW), string-matching, and other efficient time series alignment methods within the IR community. All such alignment procedures have complexity $O(NK)$, where N and K are the lengthes of the two sequences. Accordingly, it is desirable to keep the length, or dimension, of the representation as small as possible. However, this often comes at the expense of retrieval performance.

In the present work we compare the relative merits of three time series representations for sung queries. Two representations have been previously proposed: a pitch contour [1] [16, 28], and a sequences of notes [2, 6, 17, 20]. A unified presentation of the alignment procedure is given for the two representations. For the contour representation we explore multiple continuity constraints and find that judicious slope constraints improve performance considerably. We also propose a novel sequence of pitch histograms derived from the pitch contour. Unlike other pitch histogram representations [9, 24], which were proposed to allow for errors in pitch detection, we employ pitch histograms to eliminate ambiguous timing information, thus quickening the alignment procedure. This representation violates the local comparison constraint of the common DTW algorithm, hence a modified dynamic alignment procedure is presented.

We evaluate the performance of the three representations in the context of a query-by-humming (QBH) system. Early QBH successes [11, 17] have suggested numerous alternative techniques [6, 9, 20], which upon further investigation have yielded results that are often inconclusive and difficult to generalize [6, 7]. In the present work we focus on relative performance trends between the three representations rather than the absolute performance of any one. We apply our methods to a variable database of themes, similar to [6]. In so doing, we observe some interesting trends between the different methods.

As for any experiment, numerous simplifications are made to facilitate measurement and eliminate distracting details. Our hope is that, by considering how the performance decreases as target database size increases, we arrive at a measure of how the methods would perform in a broader context. That is, many of the complicating factors that affect QBH performance in a real-world scenario can be addressed by expanding the size of the target database [6].

The next three sections describe, in turn, the computation and alignment of the contour representation, the note representation and the histogram representation. Section 5

[1] In the present work, the "pitch contour" is defined as the output of a pitch tracking algorithm. See Fig. 1 for a sample pitch contour. Other authors use "pitch contour" to refer to a coarsely quantized sequence of pitch differences of a note sequence, as in Parson's directory of themes, i.e., "UDUDRU" for example [19].

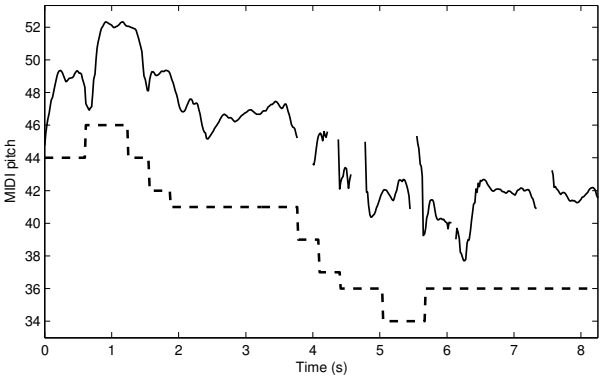

Figure 1. Sample query and target pitch contours for the main theme from Richard Rodgers' "Sound of Music." The query pitch contour is given by the solid line and the target pitch contour is given by the dashed line.

presents our methods for evaluation, along with results and discussion. Section 6 concludes the paper.

2. CONTOUR REPRESENTATION

The present work is predicated on the assumption that queries should be coded using primarily pitch information. Singers can sing the same melody with varying amplitude envelopes, lyrics and style; we desire our query representation to be invariant to such variables, however. As such, all three query representations considered here are derived from the pitch contour. The pitch contour is estimated using a time-domain method [2, 4]. This algorithm computes the autocorrelation for overlapping windows of recorded data. The bias of the window function is mitigated by the normalization $\tilde{r}(\tau) = \frac{r(\tau)}{r_w(\tau)}$, where $r(\tau)$ is the autocorrelation of the windowed data and $r_w(\tau)$ is the autocorrelation of the window function. A set of candidate peaks is selected for every frame and the Viterbi algorithm is used to construct a smooth contour. We use a step-size of 10 ms throughout this work. The fundamental frequency values are then converted to MIDI pitch [2], yielding the final pitch contour. Fig. 1 shows a sample pitch contour for the main theme from "Sound of Music." Included in Fig. 1 is the piecewise constant ideal contour for the same theme stored in our target database. Note that in this figure the target contour has been time-scaled to have equal duration as the query. Furthermore, note that the query contour contains gaps corresponding to short pauses taken by the singer; the pitch track algorithm performs an implicit voiced/ unvoiced segmentation.

The query pitch contour can either be used directly by the retrieval system, as proposed in [15, 16, 28], or further coded. In the present work we extend the alignment procedure presented in [15, 16] and observe some interesting trends relative to other query representations. The remainder of this section describes the alignment procedure for the contour representation.

[2] The real-valued MIDI pitch number p is related to a signal's fundamental frequency in Hz, f_0, as $p = 12\log_2(f_0/261) + 60$.

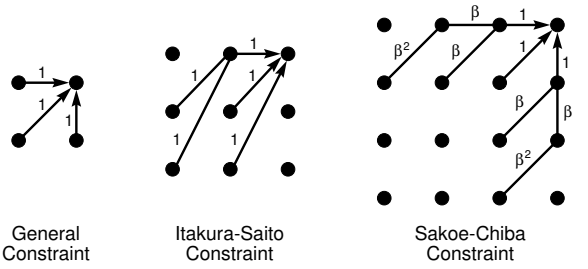

Figure 2. Three local continuity constraints for DTW. The Sakoe-Chiba continuity constraint is shown with a slope constraint of $p = \frac{1}{2}$ [23].

The retrieval component compares the query pitch contour with the piecewise constant contour for every theme stored in the target database. Let the query pitch contour be given by $\mathbf{Q} = (q_1, q_2, \cdots q_K)$ and the target contour given by $\mathbf{T} = (t_1, t_2, \cdots t_N)$. A direct, albeit 'brittle,' dissimilarity measure is given by [14]

$$D = \sum_{k=1}^{\min(K,N)} |q_k - t_k|^p. \quad (1)$$

A more robust metric is yielded with dynamic time warping (DTW), an alignment procedure popularized in the 1970's for speech recognition [10, 22, 23]. The increasing number of multimedia applications for large collections of time series has recently reinvigorated interest in DTW within the database community [14, 21, 26–28].

DTW is used to find a minimum cost alignment between the query and target contours. Let $\mathbf{\Gamma} = [\gamma_{n,k}]$ be an $N \times K$ matrix of minimum prefix alignment costs; $\gamma_{n,k}$ is the minimum alignment cost for $(q_1 \cdots q_k)$ and $(t_1 \cdots t_n)$. Let a warping path be given by $\mathbf{w} = (w_1, w_2, \cdots w_T)$, where $w_t = (n, k)$ indicates that q_k is aligned with t_n. The warping path must adhere to several constraints to be physically meaningful. The path must begin with the first elements of the query and target contour, and end with the last elements of the query and target; $w_1 = (1, 1)$ and $w_T = (N, K)$. The path must be monotonic nondecreasing and adhere to a local continuity constraint. Numerous continuity constraints have been explored [10, 21–23, 26]. In the present work we consider three common continuity constraints; the general constraint [15, 22], the 'Itakura' constraint [10] and the 'Sakoe' constraint [23]. The Itakura and Sakoe continuity constraints place bounds on the slope of the alignment path. The three continuity constraints are shown in Fig. 2. Starting in the lower left corner of $\mathbf{\Gamma}$, every element of $\mathbf{\Gamma}$ is found recursively by

$$\gamma_{n,k} = \min \begin{pmatrix} \gamma_{n-1,k} \\ \gamma_{n,k-1} \\ \gamma_{n-1,k-1} \end{pmatrix} + \text{Match Cost}(q_k, t_n) \quad (2)$$

for the general continuity constraint, where

$$\text{Match Cost}(q_k, t_n) = |q_k - t_n|^p. \quad (3)$$

The recursive equations for $\gamma_{n,k}$ for the Itakura and Sakoe continuity constraints are similar [1, 23, 26], with edge weights shown in Fig. 2. $\beta \geq 1$ is an extra cost penalty applied to favor more direct alignment paths. To speed computation and prevent pathological warpings, the alignment path is not allowed to stray too far from the diagonal $n = k$. That is, $\forall\, n, k:\ |n-k| > r \rightarrow \gamma_{n,k} = \infty$, where r is the window length, or radius, of the alignment path.

The final alignment cost for $\mathbf{Q}$ and $\mathbf{T}$ is given by $\gamma_{N,K}$, and is interpreted as a dissimilarity measure. By performing this alignment on every target in the database we can rank order the target themes. The complexity of this alignment procedure is $O(NK)$. Note that the final alignment cost is *not* normalized by the total length of the warping path. Using $\gamma_{N,K}/T$ as the final metric is common in DTW systems to avoid penalizing long targets [15,22,23]. However, as discussed below, our system time-scales all targets to the length of the query prior to alignment. Normalizing by the total warping path length prohibits the use of many cost schemes, as the DP algorithm is no longer guaranteed to find the optimal alignment path after normalization. The cost scheme we use for the Sakoe continuity constraint, for example, violates the DP constraint if the final alignment costs are normalized. As such, we do not normalize the final alignment costs. However, whether the use of a cost scheme that violates the DP constraint (in which case the final alignment path is suboptimal) detriments *retrieval* performance is an unanswered question.

2.1. Implementation issues

As can be seen in Fig. 1, the query pitch contour contains gaps wherever the singer is not articulating a coherent pitch. It is unclear how to match such 'null' query regions to the target contour. Accordingly, null query regions are filled in with a constant pitch, computed as the weighted average of the last few contour values of the previous real-valued region.

Queries are rarely sung in the same key, or with the same tempo, as the target theme. To achieve tempo invariance, the target note durations are scaled to yield a contour of total length equal to the query contour. To achieve transposition invariance, the average pitch difference between the query and target is removed prior to alignment. Note that these two normalizations achieve the desired invariance only if the two contours have approximately the same shape; as discussed in Sec. 5.1, we assume the user sings the same part of the theme as is stored in the target database.

Direct use of the pitch contour yields a query representation of relatively large dimension. For a step-size of 10 ms, typical pitch contours are of length $\sim 10^3$, whereas the note sequences discussed in the next section are of length $\sim 10^1$. Direct use of the contour proved to be computationally burdensome. Accordingly, we employ a simple dimension reduction technique proposed in [14]; query pitch contours are decimated by a factor of 10, yielding a representation of length $\sim 10^2$ [15].

3. NOTE REPRESENTATION

A musically intuitive query representation is a sequence of (pitch, duration) pairs, e.g., an estimate of the note sequence sung by the user. In this case the dimension of the query representation is relatively small, on the order of 10^1.

In [2, 3] several sung melody transcription methods are explored for use in QBH systems. Two of the methods discussed in [2] are included in the present study: the best and worst performers. The primary difference between the two note estimators is the note segmentation. The poor note estimator uses a smoothed derivative of the pitch contour to detect note boundaries [6, 17]. The good note estimator uses an HMM with 12 pitch states per octave and finds the most likely state sequence for the observed contour. The two note estimators are included in the present study to indicate the range of performance possible using a sequence of notes to represent a sung query. Because note off-set time is an unreliable statistic, inter-onset interval (IOI) is used for note duration [18].

A common method for comparing two note sequences is taken from biological sequence analysis [8]. Let the query sequence of K notes be given by $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \cdots \mathbf{q}_K)$, where $\mathbf{q}_k = (q_{k,\text{pit}}, q_{k,\text{dur}})$, and let the target sequence of N notes be given by $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \cdots \mathbf{t}_N)$. An alignment between the two sequences is achieved by inserting, deleting, and replacing elements of the query in order to match the target. Numerous cost schemes were explored, and we found a simple approximation to an edit-distance DTW metric to be particularly effective [1]. Inserting or deleting a note yields a cost equal to the duration of the note. Replacing $\mathbf{q}_k$ with $\mathbf{t}_n$ has cost

$$\text{Replace Cost}(\mathbf{q}_k, \mathbf{t}_n) = |q_{k,\text{dur}} - t_{n,\text{dur}}| + \frac{2}{\alpha}\min(q_{k,\text{dur}}, t_{n,\text{dur}})|q_{k,\text{pit}} - t_{n,\text{pit}}|. \quad (4)$$

The replacement cost has two components, a cost proportional to the difference in durations (to make the durations equal), and a cost proportional to the pitch difference times the minimum of the two durations (to make the pitches equal, either before insertion or after deletion). α is the "cross-over" pitch difference that relates the replacement cost with insertion and deletion costs. Replacing two equal-duration notes with a pitch difference equal to α has equal cost to an insertion-deletion pair.

Similar to DTW, an optimal alignment between $\mathbf{Q}$ and $\mathbf{T}$ is found by recursively building a matrix $\mathbf{\Gamma} = [\gamma_{n,k}]$ of minimum prefix alignment costs. Let the alignment path be given by $\mathbf{w} = (w_1, w_2, \cdots w_T)$. The path must be monotonic nondecreasing and adhere to the general local continuity constraint; $\forall t,\ w_t - w_{t-1} \in \{(0,1), (1,1), (1,0)\}$. Starting in the lower left corner of $\mathbf{\Gamma}$, every element of $\mathbf{\Gamma}$ is found recursively by

$$\gamma_{n,k} = \min \begin{pmatrix} \gamma_{n-1,k} + \text{Insert Cost}(\mathbf{t}_n), \\ \gamma_{n,k-1} + \text{Delete Cost}(\mathbf{q}_k), \\ \gamma_{n-1,k-1} + \text{Replace Cost}(\mathbf{q}_k, \mathbf{t}_n) \end{pmatrix}. \quad (5)$$

The final alignment cost for $\mathbf{Q}$ and $\mathbf{T}$ is given by $\gamma_{N,K}$. The complexity of this alignment procedure is $O(NK)$.

To achieve tempo invariance, all query and target note durations are normalized by the mean query and target note duration, respectively. To ensure transposition invariance, we iteratively subtract the mean pitch difference between the aligned sequences and then realign the sequences.

4. HISTOGRAM REPRESENTATION

As will be shown in the next section, the contour representation yields more robust performance than the note representation. This comes at the cost of a much larger query representation, however. Indeed, for our MATLAB implementation, aligning a pair of note sequences takes about 0.001s and aligning a pair of pitch contours takes about 0.1s, similar to [6]. The contour representation is impractical for any real-world QBH system with a large target database. This observation motivates an alternative representation that yields similarly robust performance as the contour representation, but without the computational burden. We propose a novel sequence of pitch histograms.

Perhaps the most difficult component of sung query transcription is note segmentation [2,3,17,18,28]. There is an inherent tradeoff between insertion and deletion errors in any segmentation problem [2, 12]. Most QBH systems are implicitly tuned to have roughly equal note insertion and deletion rates. An interesting exception is presented in [21], in which a note estimator with a high insertion rate is coupled with an alignment procedure that accounts for many insertion errors. As discussed in section 2, the pitch detection algorithm employed in the present work performs a partial segmentation, which can be interpreted as a note segmenter with a high note deletion rate [3] . In this case, each contour region represents one or more notes. Each region is collapsed into a single histogram of pitches. In so doing, we model the sung query as a partially ordered set [4] .

Pitch histograms have been proposed before for MIR. Tzanetakis and Cook [25] proposed pitch histograms for genre classification and Heo et. al. [9] and Song et. al. [24] have proposed sequences of pitch histograms for query-by-humming. There are some important distinctions however. In [9, 24] pitch histograms are employed to account for uncertainty in pitch detection as well as polyphonic sources. In contrast, we employ pitch histograms to discard ambiguous timing information within each contour region. Furthermore, [9,24] compute pitch histograms for a constant frame size, whereas we compute a single histogram for every contour region. In this case the duration represented by each histogram varies, and a modified DP alignment algorithm is required.

[3] Indeed, applying such a note segmenter to our database of sung queries (that is, using the gaps in the pitch contour to indicate new notes) yields a segmentation with a deletion rate of about 35% [2].

[4] The order of contour regions is defined, but the order of pitches within each region is not.

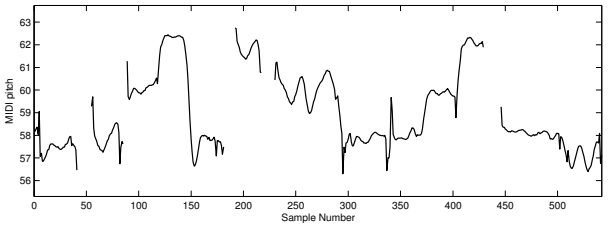

Figure 3. Sample query contour for "Yankee Doodle."

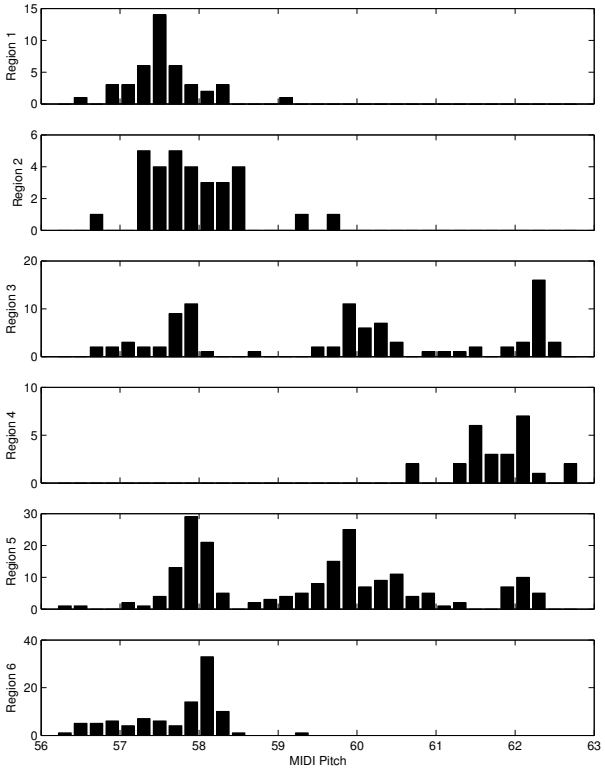

Figure 4. Pitch histograms for contour in Fig. 3.

4.1. Query & target histograms

A sample query pitch-contour for the tune "Yankee Doodle" is shown in Figure 3, and the corresponding sequence of pitch histograms is shown in Figure 4.

Let the sequence of query histograms be given by $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_K)$, and the target sequence given by $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N)$. Each query histogram is given by $\mathbf{q}_k = [q_k^1, q_k^2, \cdots, q_k^M]$, and every target histogram by $\mathbf{t}_n = [t_n^1, t_n^2, \cdots, t_n^M]$, where the number of pitch contour values in the m^{th} bin of the k^{th} query histogram is given by q_k^m. Both sequences are normalized to unit 'volume,'

$$\sum_{k=1}^{K}\sum_{m=1}^{M} q_k^m = \sum_{n=1}^{N}\sum_{m=1}^{M} t_n^m = 1, \qquad (6)$$

thus guaranteeing tempo-invariance. To make the representation transposition-invariant, the mean pitch difference between the query and target *contours* is subtracted

from the contours prior to computing the histograms, as described in Section 2.1.

The target sequence is represented using a separate histogram for every note; $\forall\, n,\ \exists!\ m$ S.T. $t_n^m > 0$. Users often sing portions of a melody continuously, hence typically $K < N$. Because $\mathbf{q}_k$ may represent more than one note, it is necessary to compare $\mathbf{q}_k$ to a collection of target histograms. Accordingly, let $\mathbf{t}_{(n,p)}$ be the cumulative sum of $\mathbf{t}_n$ through $\mathbf{t}_p$.

4.2. Match cost

We use a musically intuitive match cost for comparing $\mathbf{q}_k$ and $\mathbf{t}_{(n,p)}$ that shares some features with quantization error. The match cost is computed by partitioning $\mathbf{q}_k$ into $p - n + 1$ cells; each query cell is then 'quantized' to the corresponding non-zero bin in $\mathbf{t}_{(n,p)}$. A Voronoi partition is used to define cell boundaries [5]. The cost of each cell is given by the sum of a 'duration' component and a 'quantization' component. The duration component is the difference between the total duration represented by the query and target cells. The quantization component is given by the absolute error incurred by 'quantizing' the query bins to the non-zero bin in the target cell. A detailed description of the cell cost can be found in [1]. The final match cost is given by the sum of all cell costs.

4.3. Alignment procedure

For the contour representation, every query pitch is matched to one of N possible target pitches. For the histogram representation however, every query histogram, $\mathbf{q}_k$, is matched to one of $\frac{N^2+N}{2}$ possible cumulative target histograms, $\mathbf{t}_{(n,p)}$. To find the optimal alignment between sequences $\mathbf{Q}$ and $\mathbf{T}$ we construct a $N \times N \times K$ matrix $\mathbf{\Gamma} = [\gamma_{n,p,k}]$. The $(n,p,k)^{\text{th}}$ element of $\mathbf{\Gamma}$ represents the minimum alignment cost for the prefix subsequences $(\mathbf{q}_1 \cdots \mathbf{q}_k)$ and $(\mathbf{t}_1 \cdots \mathbf{t}_p)$ with $\mathbf{q}_k$ matched to $\mathbf{t}_{(n,p)}$. Note the critical assumption, that every histogram aligns with at least one target histogram. Starting with $k = n = 1$, every element of $\mathbf{\Gamma}$ is found recursively by [1]

$$\gamma_{n,p,k} = \min_r\ \gamma_{r,n-1,k-1} + \text{Match Cost}(\mathbf{q}_k, \mathbf{t}_{(n,p)}). \quad (7)$$

Figure 5 shows an example of the points in $\mathbf{\Gamma}$ that the alignment path can visit. This figure shows the case for $N = 12$ and $K = 6$. The dark squares in the k^{th} panel of the figure show all the possible cumulative target histograms that the k^{th} query histogram can be matched to. For each panel, the vertical index gives the starting point n for the cumulative histogram and the horizontal index gives the ending point p. For example, the first query histogram, $k = 1$, must match to a cumulative target histogram beginning with $n = 1$, but can end anywhere from $p = 1$ to $p = 7$ ($p = 8 \cdots 12$ is disallowed because at least five target histograms are needed to match to the remaining five query histograms).

[5] Cell boundaries are given by the midpoint between non-zero bins in $\mathbf{t}_{(n,p)}$

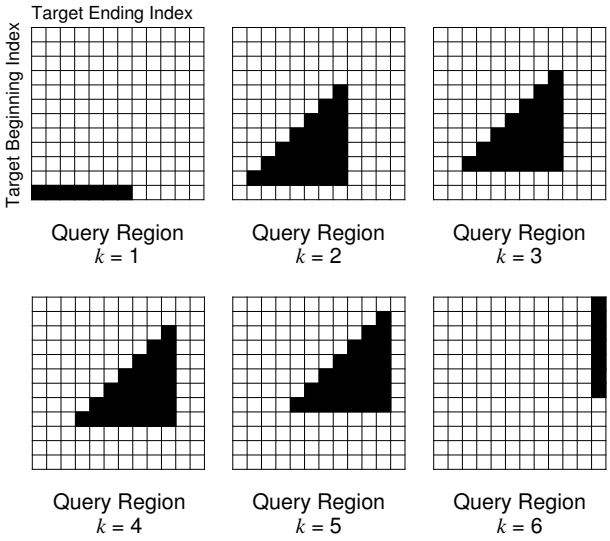

Figure 5. This figure shows the allowable alignment points in the coast matrix $\mathbf{\Gamma}$. Each panel shows one slice of the matrix, corresponding to one region of query pitch contour. Dark squares represent points the alignment path may visit. In this example the query is represented by a sequence of six histograms and the target by a sequence of twelve histograms.

5. EVALUATION

5.1. Query Database

For performance evaluation, we employ a query test set containing many sample queries of fourteen popular tunes, from the Beatles' "Hey, Jude" to Richard Rodgers' "Sound of Music." A total of 480 queries were collected from fifteen participants in our study. Each participant was asked to sing a familiar portion of a subset of the fourteen tunes four times each. The participants had a variety of musical backgrounds; some had considerable musical or vocal training while most had none at all. Participants were instructed to sing each query as naturally as possible using the lyrics of the tune [6]. The queries are monophonic, 16 bit recordings sampled at 44.1 kHz and resampled to 8 kHz to reduce processing time. The queries are between 5 and 20 seconds in length. All data were collected in a quiet classroom setting and participants were free to progress at their own pace.

Note that for all fourteen melodies, every participant sang the same set of measures. For a real-world QBH system, this is an unreasonable assumption. Some systems address this problem by specifically allowing for inserted and deleted notes at the beginning and end of the theme in the retrieval component [18]. Another common practice is to include multiple themes for each tune in the target database [6], increasing the size of the target database.

[6] This contrasts substantially from the common practice of having participants sing isolated pitches on a neutral vowel, requiring participants to perform note segmentation [2].

5.2. Target Database

Measuring the retrieval performance of our QBH methods on a target database of only the fourteen themes for which we have sample queries would not indicate how well the methods would perform in a broader context. Augmenting the target database with extra themes that are not included in the query test set was proposed in [6]. In the present work we follow suit and investigate how the various QBH methods perform as the size of the target database increases.

How the target database of themes is augmented is of critical importance. The additional themes must be sufficiently similar to the original themes to substantially detriment retrieval performance. That is, augmenting our target database with themes that are very different from those represented in the test set would not affect performance. Scaling the target database to include thousands of similar 'authentic' themes is difficult. Accordingly, we augment the fourteen authentic themes with a varying number of 'synthetic' target themes. It should be noted from the outset that using synthetic targets limits how the results can be interpreted; we are concerned with relative trends rather than absolute performance, however. We generate synthetic themes using a Markov model designed to yield note sequences with similar first-order statistics as the fourteen authentic themes. For each synthetic target, two Markov models are built; one for generating a sequence of note pitches, in which every state represents a unique pitch, and one for generating a sequence of note durations, in which every state represents a unique duration. The fourteen authentic targets are used to assign state transition probabilities. A detailed description of the Markov models can be found in [1]. Note that many of the synthetic themes are musically unsatisfying, since they obey only the first-order Markov properties of the original themes. However, because none of the QBH methods we explored consider higher-order statistics of the query, the synthetic targets do not favor one method over the others.

5.3. Results

Two performance metrics are reported: the classification accuracy and mean reciprocal rank. The classification accuracy (*CA*) is the fraction of queries that are classified as the correct target. The mean reciprocal rank (*MRR*) is the average inverse rank of the correct target. Note that $CA \leq MRR \leq \frac{1+CA}{2}$. Using the query and target databases described above, the *CA* and *MRR* is computed for the seven QBH implementations presented in Sections 2 through 4. Four implementations using the contour representation are included: the direct comparison of (1) as well as DTW using the general, Itakura and Sakoe continuity constraints. Two implementations using the note representation are included: the pitch-derivative note estimator (labelled "Note: ΔP Seg.") and the HMM note estimators. Lastly, the histogram representation is included.

Fig. 6 displays the classification accuracy for the 7 QBH methods. The target database size is represented along the abscissa and the classification accuracy along the ordinate. The fourteen 'authentic' themes are included in every database size, hence for the largest target database size, 3570 of the 3584 themes are 'synthetic.' For all but the smallest and largest target databases, the points shown are an average across multiple target databases. Data are shown along with best-fitting linear curves [7]. Clearly, the linear fit cannot be extrapolated indefinitely, *CA* cannot become negative. Nonetheless, a linear fit provides a pragmatic visual aid and implies a 'slope,' or rate of performance degradation. Fig. 7 displays the mean reciprocal rank for the same 7 QBH systems across the same range of target databases. For both figures, the contour representations are shown with a solid line, the note representations with a dashed line, and the histogram representation with a dash-dot line.

5.4. Discussion

In Fig. 6, the relative performance of the various QBH methods demonstrate several interesting trends. As expected, direct comparison of pitch contours, without alignment, yields the poorest performance. It is striking however that using a common note estimator *with* alignment only yields marginally better performance, and that this improvement quickly vanishes as target database size increases. That is, melody transcription coupled with alignment does not necessarily perform any better than Euclidean distance applied directly to the pitch contours.

The HMM note estimator yields considerably better performance than the simple pitch derivative. Indeed, for a small target database size, the note representation computed using the HMM estimator yields the best performance, $CA \approx 92\%$. However, the rate of performance degradation for this method is considerably faster than that of the best contour representations. The contour representation using the Itakura and Sakoe continuity constraints yield the most robust performance, the Sakoe constraint in particular. For the largest target database size the best contour representations yield $CA \approx 80\%$ whereas for the best note representations $CA \approx 75\%$. That the contour representation is more robust to increasing target database size is not surprising; as the number of targets grows, targets placed in ~ 10-D space will inevitably be closer than targets in ~ 100-D space.

The histogram representation does not outperform the best note or contour representations. However, the rate of performance degradation for the histogram representation is considerably slower than that of the note representations. The *CA* slope for the histogram representation is equal to that of the contour representation using the Itakura continuity constraint. For the largest target database, the histogram representation yields equal *CA* as the best note representation. This is an intriguing observation because while the contour representation yields the most robust performance, it is computationally burden-

[7] This is in contrast to [6], who found a $\frac{1}{x}$ fit (on a log-scale) to better match their data.

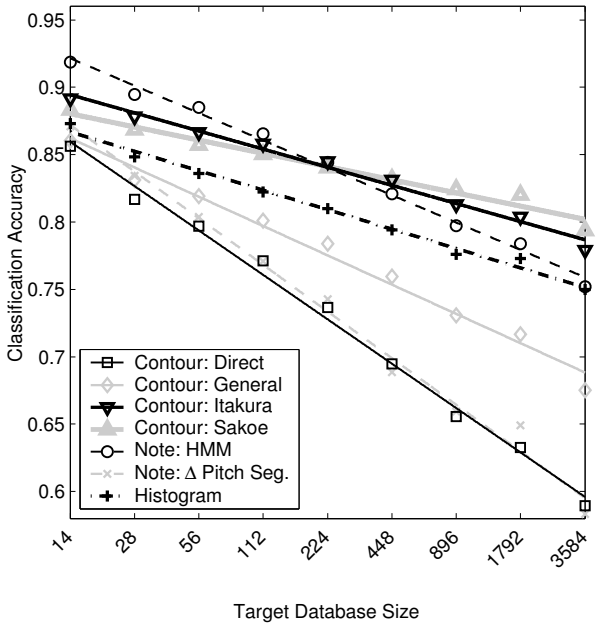

Figure 6. Classification accuracy for 8 QBH methods. The independent variable is the size of the target database. For every target database, the 14 themes represented in the query test set are included.

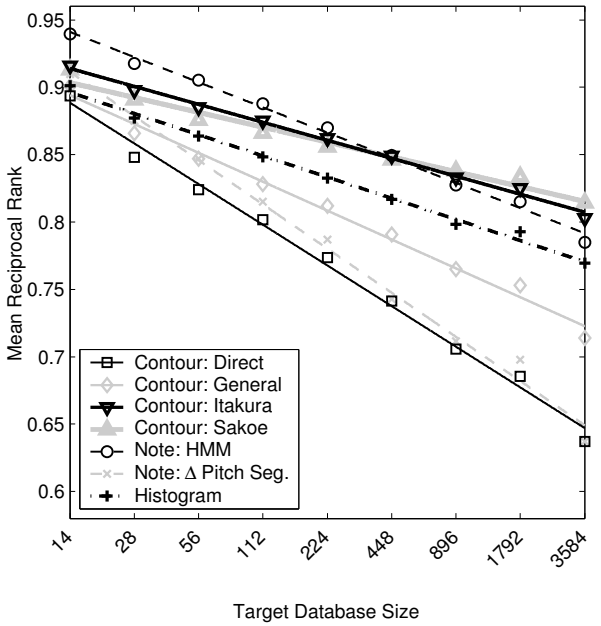

Figure 7. Mean reciprocal rank for 8 QBH methods.

some. Alignment of histogram sequences is only modestly more time consuming than for the note representation [8] . As the target database is scaled to massive proportions, the histogram representation could provide the best compromise between performance and retrieval speed.

The performance of the histogram representation could be improved by more judicious 'continuity' constraints and histogram match costs. For the contour representation, the best performance is achieved with the Itakura and Sakoe constraints. No alternative constraints were explored for the histogram representation, there being no obvious interpretation for the 'slope' of the alignment path. Furthermore, the alignment procedure assumes that every query histogram represents at least one target note, insertion errors are disallowed. Some of the query pitch contours contain spurious contour regions which result from environmental noise, however. Many of these spurious regions could be discarded using a minimum duration constraint, but those that remain cause the alignment procedure to find implausible alignments.

Comparing Figures 6 & 7, the note representations yield relatively better performance in terms of *MRR* than *CA*. Indeed, for the largest target database, the median rank of all misclassified queries using the note representation is about 30, whereas the median rank is about 100 for the contour representation. This is due to the different cost schemes. The contour and histogram cost schemes do not allow for portions of the query to be explicitly 'deleted' or 'inserted,' every element of the query must be matched to at least one element in the target. This occasionally results in pathological alignments, radically warping the query sequence to account for a short portion of incongruous data. In such situations the note representation alignment simply deletes or inserts the appropriate element.

For the target database of 14 themes, the best classification accuracy observed is about 92%. We note that classification accuracy of 98% is achieved by removing the sample queries of three subjects from our test database. These queries were very inaccurate and some were virtually monotone; the lyrics were their only recognizable feature. It is unclear that any QBH system should be designed to accommodate such queries. Specific modeling of singer and transcription error is becoming a more active area of QBH research [18, 21]. As mentioned at the outset, our query representations are based exclusively on the estimated pitch contour. Employing other features such as broad spectral or phonetic information may further improve performance.

6. CONCLUSION

In this work we have explored several implementations of three different query representations: a sequence of notes, a pitch contour (a sequence of uniformly sampled pitch estimates), and a sequence of pitch histograms. We found that the note representation lends itself to the most rapid alignment yet the least robust performance. Conversely, the contour representation lends itself to the slowest alignment but the most robust performance. The histogram representation yields a compromise between the note and contour representations.

[8] For our MATLAB implementation, aligning a pair of histogram sequences takes about 0.003s. Indeed, for the histogram representation, $\mathbf{\Gamma}$ contains $\sim 10^3$ elements. Whereas for the note and contour representations, $\mathbf{\Gamma}$ contains $\sim 10^2$ and $\sim 10^4$ elements, respectively.

7. REFERENCES

[1] N.H. Adams, *Time Series Representations for Music Information Retrieval*, University of Michigan (2004), http://www.eecs.umich.edu/techreports/systems/cspl/cspl-349.ps.gz.

[2] N.H. Adams, M.A. Bartch, and G.H. Wakefield, *Note Segmentation and Quantization for Music Information Retreival*, *submitted to* IEEE Transactions on Speech and Audio Processing.

[3] ———, *Coding of Sung Queries for Music Information Retreival*, IEEE Workshop on Application of Signal Processing to Audio and Acoustics (2003).

[4] Paul Boersma, *Accurate Short-Term Analysis of the Fundamental Frequency and the Harmonics-to-Noise Ratio of a Sampled Sound*, Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam, vol. 17, 1993, pp. 97–110.

[5] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, *Robust Sound Modeling for Song Identification in Broadcast Audio*, AES Convention (Munich, Germany), May 2002.

[6] R. B. Dannenberg, W. P. Birmingham, and G. Tzanetakis et. al., *The MUSART Testbed for Query-By-Humming Evaluation*, Proceedings of ISMIR, 2003.

[7] J. S. Downie, *Toward the Scientific Evaluation of Music Information Retrieval Systems*, Proceedings of ISMIR, 2003, Baltimore, MD.

[8] D. Gusfield, *Algorithms on Strings, Trees and Sequences*, Cambridge University Press, Cambridge, UK, 1999.

[9] S. P. Heo, M. Suzuki, A. Ito, and S. Makino, *Three Dimensional Continuous DP Algorithm for Multiple Pitch Candidates in Music Information Retrieval System*, Proceedings of ISMIR, 2003, Baltimore.

[10] F. Itakura, *Minimum Prediction Residual Principle Applied to Speech Recognition*, IEEE Transactions on Acoustics, Speech and Signal Processing **23** (1975), no. 1, 67–72.

[11] T. Kageyama, K. Mochizuki, and Y. Takashima, *Melody Retrieval with Humming*, Proceedings of Int. Computer Music Conference (ICMC), 1993, Tokyo.

[12] S. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, Prentice Hall Ptr, Upper Saddle River, NJ, 1998.

[13] E. Keogh, *Exact Indexing of Dynamic Time Warping*, Proceedings of the 28th VLDB Conference (2002).

[14] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, *Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases*, Knowledge and Information Systems 3(3) (2000).

[15] D. Mazzoni, *Tech Report: Melody Matching Using Time Warping*, Technical Report, Carnegie Mellon University (2002).

[16] D. Mazzoni and R. B. Dannenberg, *Melody Matching Directly from Audio*, Proceedings of ISMIR 2001 (2001), Bloomington, IN.

[17] R. J. McNab, L. A. Smith, and I. H. Witten et al, *Towards the Digital Music Library: Tune Retrieval from Acoustic Input*, Proceedings of ACM Digital Libraries Conference (1996), Bethesda, MD.

[18] C. Meek and W. Birmingham, *Johnny can't sing: A comprehensive error model for sung music queries*, Proceedings of ISMIR (2002).

[19] D. Parsons, *The Directory of Tunes*, Spencer Brown and Co., Cambridge, England, 1975.

[20] S. Pauws, *Cubyhum: A Fully Operational Query by Humming System*, Proceedings of ISMIR, 2002, Paris, France, pp. 187–196.

[21] A. Pikrakis, S. Theodoridis, and D. Kamarotos, *Recognition of Isolated Musical Patterns Using Context Dependent Dynamic Time Warping*, IEEE Transactions on Speech and Audio Processing **11** (2003), no. 3, 175–183.

[22] L.R. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Upper Saddle River, NJ, 1993.

[23] H. Sakoe and S. Chiba, *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*, IEEE Transactions on Acoustics, Speech and Signal Processing **26** (1978), no. 1, 43–49.

[24] J. Song, S.Y. Bae, and K. Yoon, *Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System*, Proceedings of ISMIR (2002), Paris, France.

[25] G. Tzanetakis, A. Ermolinskyi, and P. Cook, *Pitch Histograms in Audio and Symbolic Music Information Retrieval*, Proceedings of ISMIR, 2002, Paris, France.

[26] R. Yaniv and D. Burshtein, *An Enhanced Dynamic Time Warping Model for Improved Estimation of DTW Parameters*, IEEE Transactions on Speech and Audio Processing **11** (2003), no. 3, 216–228.

[27] B.K. Yi, H. Jagadish, and C. Faloustsos, *Efficient Retrieval of Similar Time Sequences Under Time Warping*, Proc. IEEE International Conference on Data Engineering, 1998, pp. 201–208.

[28] Y. Zhu and D. Shasha, *Warping Indexes with Envelope Transforms for Query by Humming*, Proc. International Conference of Management of Data (SIGMOD) (San Diego, CA), 2003.