EFFICIENT MELODIC QUERY BASED AUDIO SEARCH FOR HINDUSTANI VOCAL COMPOSITIONS

Kaustuv Kanti Ganguli[1] **Abhinav Rastogi**[2] **Vedhas Pandit**[1]
Prithvi Kantan[1] **Preeti Rao**[1]
[1] Department of Electrical Engineering, Indian Institute of Technology Bombay
[2] Electrical Engineering, Stanford University
kaustuvkanti@ee.iitb.ac.in

ABSTRACT

Time-series pattern matching methods that incorporate time warping have recently been used with varying degrees of success on tasks of search and discovery of melodic phrases from audio for Indian classical vocal music. While these methods perform effectively due to the minimal assumptions they place on the nature of the sampled pitch temporal trajectories, their practical applicability to retrieval tasks on real-world databases is seriously limited by their prohibitively large computational complexity. While dimensionality reduction of the time-series to discrete symbol strings is a standard approach that can exploit computational gains from the data compression as well as the availability of efficient string matching algorithms, the compressed representation of the pitch time series itself is not well understood given the pervasiveness of pitch inflections in the melodic shape of the raga phrases. We propose methods that are informed by domain knowledge to design the representation and to optimize parameter settings for the subsequent string matching algorithm. The methods are evaluated in the context of an audio query based search for Hindustani vocal compositions in audio recordings via the mukhda (refrain of the song). We present results that demonstrate performance close to that achieved by time-series matching but at orders of magnitude reduction in complexity.

1. INTRODUCTION

A bandish, or composition in the North Indian classical vocal genre of khayal, is characterised by its mukhda, its almost cyclically repeated refrain. The singer elaborates within the raga framework in each rhythmic cycle before returning to the main phrase of the bandish (i.e. its mukhda). The automatic detection of this repetitive phrase, or motif, from the audio signal would contribute to important metadata concerning the identity of the bandish. The mukhda is recognised by the lyrics, location in the cycle and its melodic shape. While these are in order of decreasing ease in terms of manual segmentation of the mukhda, the melodic shape characterized by a pitch contour segment is most amenable to pattern matching methods. The challenge here arises from the improvisatory nature of the genre where the raga grammar allows for considerable variation in the melodic shape of any prescribed phrase. Previous work has shown that the variability in the mukhda across the concert, similar to that of other raga-characteristic phrases in a performance, can be characterized as globally constrained non-linear time-warping where the constraint appears to depend on certain characteristics of the underlying melodic shape [16, 17, 21]. A dynamic time-warping (DTW) distance measure was used on the time-series segments to model melodic similarity under local and global constraints that were learned from a raga-specific corpus [17]. More recent work has also validated the DTW based similarity measure in the context of melodic motif discovery but the high computational costs associated with time-series search limited its applicability [3, 9, 14]. Given that DTW based local matching, with relatively minimal assumptions, on the pitch time-series derived from the audio is largely successful in modeling the relevant melodic variations, we focus on targeting similar performance with greatly reduced complexity. Computationally efficient methods to search and localize occurrences of the mukhda in a concert, given an isolated audio query phrase, have the following potential real-world applications: (i) automatic segmentation of all occurrences of the mukhda provided one manually identified instance, with a goal to reduce manual effort in the rich transcription of concert audio recordings, and (ii) retrieving a specific bandish from a database of concert recordings by querying by its mukhda provided either by an audio fragment or by user singing.

The acoustic correlate of the melodic shape of a phrase is its pitch contour represented computationally by the detected pitch of the singing voice at close uniformly spaced intervals. Considering the concert recording context where an instrumental ensemble accompanies the vocalist, the pitch detection is achieved by a singing voice detection algorithm coupled with predominant F0 extraction at uniform closely spaced intervals throughout the concert. The pitch contour can be treated as a one-dimensional time-series which can be searched for the occurrence of a specific pattern as defined by the query (another time-series segment). We note that the dimensionality of the time-series is typically very high due to the required dense sampling of the pitch contour across the concert duration. It has been observed that a sampling interval on the order of 20 ms is necessary in order to preserve important pitch nuances as determined by the curve of rapidly decreasing correlation between melodically similar pitch contours with increasing sampling interval [9].

As mentioned earlier, DTW can be used in an exhaustive search across the concert of this sampled pitch time series to find the optimal cost alignment between the query and target pitch contours at every candidate location. We see therefore that any significant computational complexity reduction can only come from the reduction of dimensionality of the search space. An obvious choice is a representation of the melodic contour that uses compact musical abstractions such as a sequence of discrete pitch scale intervals (essentially, the note sequence corresponding to the melody if there was one). String-matching algorithms can then be applied that find the approximate longest common subsequence between the query and target segments of discrete symbols. Krannenburg [11] used this approach on audio recordings of folk songs to establish similarity in tunes across songs. Each detected pitch value was replaced by its MIDI symbol and the Smith-Waterman local sequence alignment algorithm was used on the resulting strings. Note however that there was no reduction in the size of the pitch time-series. If the pitch time-series is segmented into discrete notes, a far more compact string representation can be obtained by using each symbol to represent a tuple corresponding to a note value and duration. In this case, a number of melodic similarity methods based on the alignment of symbolic scores become available [1, 6, 11, 12, 27]. The effectiveness of this approach, of course, depends heavily on the correspondence between the salient features of the pitch contour and the symbol sequence. A specific challenge in the case of Hindustani vocal music is that it is characterized just as much by the precisely intoned raga notes as it is by the continuous pitch transitions and ornaments that contribute significantly to the raga identity, motivating a more careful consideration of the high-level abstraction [15, 18].

The main contributions of this work are (i) a study of the suitability of two distinct high-level abstractions for sequence representation in the context of our melodic phrase retrieval task, and (ii) using domain knowledge for the setting of various representation and search parameters of the systems. In the next section, we describe our test dataset of concerts with a review of musical and acoustic characteristics that are relevant to our task. This is followed by a presentation of our melodic phrase retrieval methods including approaches to the compact representation of the pitch time-series and discussion of the achievable reduction in computational complexity with respect to the baseline system. A description of the experiments follows. Finally the results are discussed with a view to providing insights on the suitability of particular approaches to specific characteristics of the test data.

2. TEST DATABASE DESCRIPTION

The dataset comprises 50 commercial CD-quality concert audio recordings by 18 eminent Hindustani vocal artists. The accompaniment consists of tanpura (drone) and tabla, along with harmonium or sarangi. The concerts have been chosen from a large corpus [23] in a deliberate manner so as to achieve considerable diversity in artists, ragas and tempo. We restrict our analysis to the vilambit (slow tempo) and madhyalaya (medium tempo) sections of these concerts for the current task. Drut (fast tempo) sections are excluded because their mukhda phrases contain a considerable amount of context-dependent variation and hence melodic similarity is not as strongly preserved. Table 1 summarises our dataset where 39 concerts are of vilambit laya and the remaining 11 are madhyalaya. The average duration of a vilambit bandish is 17 minutes and contains an average of 20-25 mukhda instances that occur once each in a rhythmic cycle.

# Song	Dur (hrs)	# GT	Dur (hrs)	Ratio	# Unique	
					Raga	Artist
50	13:13	1075	1:44	13%	34	18

Table 1. Description of the test dataset.

Manual annotation of the mukhda segments with start and end boundaries was carried out by a musician and validated by a second very experienced musician. Mukhdas are most easily identified by listening for the lyrical phrase that occurs about the first beat (sam) of the rhythmic cycle as evidenced by the accompanying tabla strokes. The mukhda is labeled together with its boundaries as detected from the onsets of the lyric syllables. These annotations serve as the ground truth (GT) for the evaluation of the different systems under test which exploit only the similarity of melodic shape to that of the audio query. The query thus could be an instance extracted from the audio track, or it could be a sung or hummed likeness of the melodic phrase generated by the user.

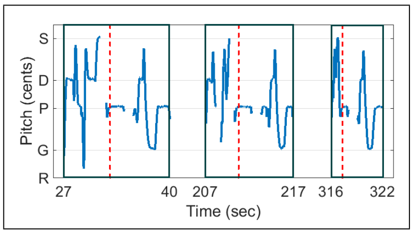

Figure 1. Pitch contour segments of distinct mukhdas. Sam of the corresponding rhythmic cycle is marked in red.

Both the cues easily available to listeners, the phones of the lyrics (as uttered by the singer) and the sam tabla strokes cannot be extracted reliably from the polyphonic audio signal. The predominant F0 extractor on the other hand is more robust and achieves the tracking of the vocalist's pitch based on dominance and continuity constraints without any explicit source separation. Our approach to mukhda detection is currently based on the computation of melodic similarity which, ideally, should encapsulate the notion of musically perceived similarity. The low-level acoustic correlate of the melody is the pitch contour, the implementation of which is presented in the next section.

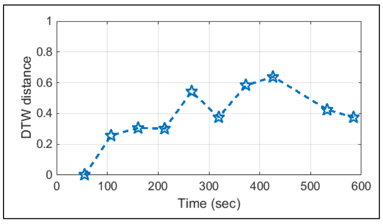

Figure 2. Normalized DTW distance between the first mukhda of the concert and subsequent mukhdas.

Figure 1 shows pitch contour segments of three mukhdas manually extracted from the beginning, middle and towards the end of the madhyalaya bandish of a concert. Also marked is the location of the sam with respect to the mukhda pitch trajectory. We note the variability in the melodic shape. Typically the tempo of the concert increases gradually over time (linked to the reduction in the rhythmic cycle duration) leading to a decrease in mukhda duration (from 13 sec to 7 sec in Figure 1). Rather than a linear compression, the melodic shape is modified by non-linear time warping [5]. Figure 2 shows a plot of DTW distance between the first mukhda of the concert and each later mukhda versus the temporal location (the corresponding sam) of the later mukhda. The distances are normalized with respect to that of the first false detection. We observe a trend of decreasing similarity with increasing time, as well as the fact that the intervals between mukhdas are not identical due to rhythmic cycle duration variability. Also, not every rhythmic cycle is marked by a mukhda. Finally, we note that the DTW distance measure is largely insensitive to the irrelevant differences, as seen from the distance values normalised with respect to the distance between the first mukhda and the nearest false detection.

3. MELODIC PHRASE RETRIEVAL SYSTEMS

In this section, we consider various approaches towards our end goal which involves searching the entire vocal pitch track extracted from the audio recording to identify pitch contour sub-segments that match the melodic shape of the query. We present the audio pre-processing required to generate the pitch time-series followed by a discussion of the different systems in terms of algorithm design and complexity.

3.1 Time series extraction from audio

The desired time-series representation is expected to capture the melody line, and hence requires accurate pitch detection of the main voice in polyphonic audio. The singing voice usually dominates over other instruments in a vocal concert performance in terms of its level and continuity over relatively large temporal extents although the accompaniment of tabla and other pitched instruments such as the drone and harmonium are present. Predominant-F0 detection is implemented by the salience based combination of two algorithms [20] which exploit the spectral properties of the voice with temporal smoothness constraints on the pitch. The pitch is detected at 20 ms intervals throughout the audio with zero pitch assigned to the detected purely instrumental regions. Next, the pitch values in Hz are converted to the cents scale by normalizing with respect the concert tonic determined by automatic tonic detection [8]. This normalization helps match a query across concerts by different artists. The final pre-processing step is to interpolate short silence regions below a threshold (80 ms which is empirically tuned in previous studies [16, 17]) indicating musically irrelevant breath pauses or unvoiced consonants by cubic spline interpolation so as to preserve the integrity of the melodic shape.

3.2 Baseline system

Our baseline method is the "subsequence DTW", an adaptation of standard DTW to allow searching for the occurrence and alignment of a given query segment within a long sequence [13, 26]. Given a query Q of length N symbols and a much longer sequence S of length M (i.e. the song or concert sequence in our context) to be searched, a dynamic programming optimization minimizes the DTW distance to Q over all possible subsequences of S. The allowed step-size conditions are chosen to constrain the warping path to within an overall compression / expansion factor of 2. No further global constraint is applied. The candidate subsequences of the song are listed in order of increasing DTW distance to which a suitable threshold can be applied to select and localize the corresponding regions in the original audio. The time complexity of subsequence DTW is $O(MN)$ where $N(M)$ is the number of pitch samples corresponding to the query (song) duration (i.e. 50 pitch samples per second of the time series duration, given that the pitch is extracted at 20 ms intervals) [2, 13, 28]. We see that the time-series dimensions contribute directly to the complexity of the search. Our goal is to find computationally simple alternatives to DTW by moving to low dimensional string search paradigms. This requires principled approaches to converting the pitch time-series to a discrete symbol sequence, two of which are presented next.

3.3 Behavior based system

With a goal to preserve the characteristic shape of the mukhda including the pitch transitions in the mapping to the symbol sequence, we consider the approach of Tanaka [25] who proposed "behavioral symbols" to capture dis-

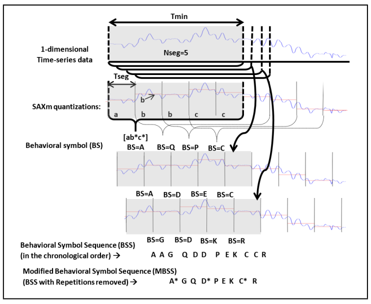

Figure 3. Construction from a pitch time series of the BS sequence (BSS) and the modified BSS.

tinct types of local temporal variation in a human motion capture system. A melodic phrase can be viewed as a sequence of musical gestures by the performer, with a behavioral symbol then potentially corresponding to a single (arbitrary movement) in pitch space. A sequence of symbols would serve as a sketch of the melodic motif. In Tanaka's system, the symbols are purely data-dependent and evolve from the analysis itself [24, 25]. We bring in musical context constraints as presented in the algorithm description next.

The pitch time-series is segmented into fixed duration windows centered at uniformly spaced intervals so that the windows are highly overlapping as illustrated in Figure 3. The pitch contour within each window is replaced by a piecewise flat contour where each piece represents a fixed fraction of the window. While Tanaka recommends normalization of the pitch values within the window to [0,1] range in order to eliminate vertical shifts and scaling between otherwise similar shapes, we omit this step given that we are not looking for transposition or scaling invariance in the mukhda detection task. The piece-wise flat subsegments are obtained by the median of the pitch values in the corresponding subsegment. We choose median as opposed to mean [24] as it is less sensitive to the occasional outliers in the pitch contour. We bring in further domain constraints by using the discrete scale intervals for the quantization of the piecewise sub-segments that describe a specific behavioral symbol (BS). We obtain a sequence of BS, one for each window position. Due to the high overlap between windows, repetitions are likely in consecutive symbols. These are replaced by a single BS which step brings in the needed time elasticity. Figure 3 illustrates the steps of construction of the BS sequence (BSS) and its repetition removed version (the modified BSS) from a simulated pitch time-series.

The database is pre-processed and the symbol sequence representation of each complete concert recording is stored. When a query is presented, it is converted to its symbol sequence (which currently depends on the song to be searched) and an exact sub-sequence search is implemented on the song string. The choice of the fixed parameters: window duration, hop duration and number of subsegments within a window turn out to heavily influence the representation. The window duration should depend on the time scale of the salient features (movements in pitch space). The subsegments must be small enough to retain the melodic shape within the window. The hop of the sliding window compensates for alignment differences of the different occurrences of the template in the pitch time-series of the song. We present "parameter settings" for two configurations.

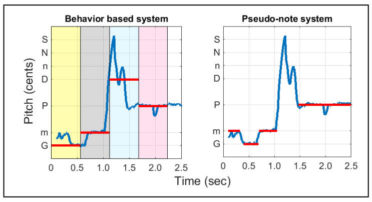

Figure 4. The two proposed systems of quantization, namely: behavior based and pseudo-note systems.

Version A: Fixed parameter setting (window = 126 samples, hop = 5 samples, # subsegments per window = 3)

Version B: Query dependent setting (window = (0.5 * N) samples, hop = 5 samples, # subsegments per window = 4)

We present next an alternate approach to symbolic representation of the pitch contour.

3.4 Pseudo-note system

An approximation to staff notation can be achieved by converting the continuous time-series to a sequence of piecewise flat segments if the section pitches are chosen from the set of discrete scale intervals of the music. If the achieved representation indeed corresponds to some underlying skeleton of the melodic shape of the phrase, we could anticipate obtaining better matches across variations of the melodic phrase. We address the question of how we can bring domain knowledge into this transformation. As we see from Figure 4, the continuous pitch contours corresponding to the phrases are not directly suggestive of a specific sequence of raga notes given that raga notes are embellished considerably when realized by the vocalist. In Indian music traditions, written notation has a purely prescriptive role and achieving the transcription of a performed phrase to written notation requires raga knowledge and much experience [19]. All the same there is a similarity across the mukhda repetitions that we wish to capture in our representation.

We consider a simple representation of the melodic shape that features only the relatively stable regions of the continuous pitch contours that lie within a musically valid interval of a scale (raga) notes. The scale notes are detected from the prominent peaks of the long-term pitch histogram across the concert and the musically valid interval is chosen to be within 35 cents [17]. This step leaves fragments of the time-series that coincide with the scale notes while omitting the remaining pitch transition regions. Next, a lower threshold duration of 80 ms is applied to the fragments to discard fragments that are considered too short to be perceptually meaningful as held notes [16]. This leaves a string of fragments each labeled by a svara (raga note as shown in Figure 4 (right)). Fragments with the same note value that are separated by gaps less than 80 ms are merged. The resulting symbol sequence thus comprises the scale notes occurring in the correct temporal order but without explicit durational information. The database is pre-processed and the symbol sequence representation of each complete concert recording is stored. When a query is presented, it is converted to its symbol sequence and an approximate sub-sequence search is implemented on the concert string based on an efficient string matching algorithm with parameter settings that are informed by domain knowledge as described next.

The similarity measurement of the query sequence with candidate subsequences of the song is based on the Smith-Waterman algorithm, widely used in bioinformatics but also applied recently to melodic note sequences [11, 22]. It performs the local alignment of two sequences to find optimal alignments using two devices. A symbol of one sequence can be aligned to a symbol of the other sequence or it can be aligned to a gap. Each of these operations has a cost that is designed as follows.

Substitution score: In its standard form, the Smith-Waterman algorithm uses a fixed positive cost for an exact match and a fixed negative score for symbol mismatch. In the context of musical pitch intervals, we would rather penalize small differences less than large differences. We present alternate substitution score functions that incorporate this.

Gap Function: This function deducts a penalty from the similarity score in the event of insertion or deletion of symbols during the alignment procedure. The default gap penalty is linear, meaning that the penalty is linearly proportional to the number of symbols that comprise the gap. Another possibility, that is more meaningful for the melody context, is the affine gap function where the gap opening cost is high compared to the cost incurred by adding each successive symbol to the gap [7]. This is achieved by a form given by $mx + c$ where x is the length of the gap and m, c are constants. Intuitively, increasing c will penalize gap openings to a greater extent, while increasing m will have a similar effect with regard to gap extension. We present different designs for the relative costs motivated by the musical context.

With variations in each of the above two controls of the Smith-Waterman algorithm, we obtain the following three distinct versions of the pseudo-note system.

Version A: This setting is similar to the default Smith-Waterman setting, with a distance-independent similarity function that assesses a score of +3 for symbol match and -1 for a substitution. Gap function is linear, with penalty equal to symbol length of gap.

Version B: Substitution score that takes pitch difference into account, i.e. Score of +3 for a match, 0 for symbols differing by upto 2 semitones, -1 for substitution, and an affine gap penalty with parameters m = 0.8, c = 1.

Version C: Query dependent settings where we use the settings of B as default with the following changes for particularly fast varying and slowly varying query melodic shapes as determined by a heuristic measure of ratio of squared number of symbols to query duration. We have the following parameter settings. (i) fast varying: Substitution score of +1 to symbols differing by upto 2 semitones. Gap penalty is affine with parameters m = 1, c = 0.5, and (ii) slowly varying: Similarity score of -0.5 to symbols differing by upto 3 semitones. Gap penalty is affine with parameters m = 0.5, c = 1.5.

Finally, the Smith-Waterman algorithm has a time complexity given by $O(MN^2)$ where N is the query length in symbols and M is the song length [22]. By constraining the allowed gap length to be no longer than that of the query itself (N), justified by the musical context, we achieve a complexity reduction to $O(MN)$.

4. EXPERIMENTS AND EVALUATION

We present experiments that allow us to compare the performance of the different systems on the task at hand, namely correctly detecting occurrences of the mukhda in the audio concert given an audio query corresponding to the melodic shape of the mukhda phrase. The queries are drawn from a set of 5 mukhdas extracted from the early part (first few cycles) of the bandish. The early mukhda repetitions tend to be of the canonical form and hence correspond well with an isolated query that a musician might generate to describe the bandish. For the investigation of a given method, we process the database to convert each concert audio to the pitch time series and then to the corresponding string representation. Next, the query is converted to the string representation and the search is executed. The detections with time-stamps are listed in order of decreasing similarity with the query as determined by the corresponding search distance measure. A detection is considered a true positive if the time series of the detection spans at least 50% of that of one of the ground-truth labeled mukhdas in the song. An ROC (precision vs recall) is obtained for each query by sweeping a threshold across the obtained distances. The ROC for a song is derived by the vertical averaging (i.e. recall fixed and precision averaged) of the ROCs of the 5 distinct queries [4]. The performance for each song is summarized by the following two measures: precision at 50% recall and the equal error rate (EER) (point on the ROC at which false acceptance rate matches false rejection rate). We further present performance of the best performing pseudo-note system on song retrieval in terms of the mean reciprocal rank (MRR) [10] on the dataset of 50 concerts as follows. We use the set of the first occurring labeled mukhda of each song to form a test set of 50 queries. Next for each test query, every song is searched to obtain a rank-ordered list of songs whose first 5 detections yield the lowest averaged distance measure to the query.

5. RESULTS AND DISCUSSION

Table 2 compares the performances of the various systems on the task of mukhda detection in terms of the average EER and average precision at a selected recall across the 50 songs where each song is queried using each of the first five mukhdas. We also report the computational complexity

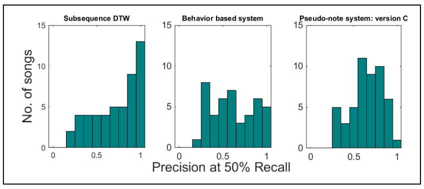

Figure 5. Histogram of the measure 'Precision at 50% Recall' across the baseline and proposed methods.

reduction factor over that of the baseline method (given by the square of the dimension reduction factor). To obtain more insight into song dependence, if any, we show the distribution of the precision values for the 50 songs set in the bar graphs of Figure 5, one system for each category, represented by the best performing one.

Method (version)		Mean	Prc at 50% Rec		Reduc.
		EER	Mean	Std.	
Subseq DTW	—	0.33	0.73	0.18	1
Behavior based system	(A)	0.47	0.56	0.26	100
	(B)	0.41	0.61	0.25	
Pseudo-note system	(A)	0.47	0.61	0.19	2500
	(B)	0.42	0.64	0.19	
	(C)	0.41	0.65	0.18	

Table 2. Comparison of the two performance measures and computational complexity reduction factor across the baseline and proposed methods.

From Table 2, we observe that the baseline system represented by subsequence DTW on the pitch time-series performs the best while the pseudo-note methods achieve the best computation time via a reduction proportional to the square of the reported dimension reduction factor (i.e. 50). We will first comment on the relative strengths of these two systems, and later discuss the behavior based system. We observe an improvement in performance of the pseudo-note system with the introduction of domain knowledge and query dependent parameter settings for the subsequence search algorithm. From Figure 5, we see that the subsequence DTW has a right-skewed distribution indicating a high retrieval accuracy for a large number of songs. However we note the presence of low performing songs too which actually do better with the pseudo-note system. Closer examination of these songs revealed that these belonged to ragas characterized by heavily ornamented phrases. In the course of improvisation, the mukhda was prefaced by rapidly oscillating pitch due to the preceding context. This led to increased DTW distance between the query and mukhda instances. The oscillating prelude was absent in the pseudo-note representation altogether leading to a better match.

The behavior based system was targeted towards capturing salient features of the melodic shape of the phrase in a symbolic representation. The salient features should ideally include steady regions as well as specific movements in pitch space that contribute to the overall melodic shape. As such, it was expected to perform better than the pseudo-note method which retains relatively sparse information as seen from a comparison of the two representations for an example phrase in Figure 4. However, the selection of the duration parameters required for the time-series conversion turned out to be crucial to the accuracy of the system. Shortening the window hop interval contributed to reduced sensitivity to time alignment differences but at the cost of reduced compression and therefore much higher time complexity. Further, the data dependence of symbol assignment requires the query to be re-encoded for every song to be searched, and further if query dependent window length is chosen, the song must be re-encoded according to the query. Future work should target obtaining a fixed dictionary of symbols to pitch movement mappings by learning on a large representative database of concerts.

Top 'M' hits	Correct songs	Accuracy
1	41 / 50	0.82
2	45 / 50	0.90
3	48 / 50	0.96

Table 3. Results of the song retrieval experiment.

Finally, we note the song retrieval performance of the pseudo-note version C in Table 3. The mean reciprocal rank (MRR) is 0.89. The top-3 ranks return 48 of the 50 songs correctly. The badly ranked songs were found to be narrowly superseded by other songs from the same raga that happened to have phrases similar to the mukhda of the true song. This suggests the potential of the method in the retrieval of "similar" songs where the commonality of raga is known to be an important factor.

In summary, the melodic phrase is a central component for audio based search for Hindustani music. Given the improvisational nature of the genre as well as the lack of standard symbolic "notation", time-series based matching of pitch contours provides a reasonable performance at the cost of complexity. The conversion to a relatively sparse representation by retaining only flat regions of the pitch contour and introducing domain driven cost functions in the string search is shown to lead to a slight reduction in retrieval accuracy while reducing complexity significantly. The inclusion of further cues such as the lyrics and rhythmic cycle markers to mukhda detection is expected to improve precision and is the subject of future research.

6. ACKNOWLEDGEMENT

This work received partial funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement 267583 (CompMusic).

7. REFERENCES

[1] N. Adams, M. Bartsch, J. Shifrin, and G. Wakefield. Time-series alignment for Music Information Retrieval. In *Proc. of Int. Soc. for Music Information Retrieval (ISMIR)*, pages 303–310, 2004.

[2] A. Chan. An analysis of pairwise sequence alignment algorithm complexities. Technical report, Stanford University, 2004.

[3] R. B. Dannenberg and N. Hu. Pattern discovery techniques for music audio. *Journal of New Music Research (JNMR)*, 32(2), 2002.

[4] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 2006.

[5] K. K. Ganguli and P. Rao. Tempo dependence of melodic shapes in Hindustani classical music. In *Proc. of Frontiers of Research on Speech and Music (FRSM)*, pages 91–95, March 2014.

[6] C. Gomez, S. Abad-Mota, and E. Ruckhaus. An analysis of the Mongeau-Sankoff algorithm for Music Information Retrieval. In *Proc. of Int. Soc. for Music Information Retrieval (ISMIR)*, pages 109–110, 2007.

[7] O. Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162:705–708, 1982.

[8] S. Gulati, A. Bellur, J. Salamon, H. G. Ranjani, V. Ishwar, H. A. Murthy, and X. Serra. Automatic tonic identification in Indian art music: Approaches and Evaluation. *Journal of New Music Research*, 43(1):53–71, 2014.

[9] S. Gulati, J. Serra, V. Ishwar, and X. Serra. Mining melodic patterns in large audio collections of Indian art music. In *Proc. of Int. Conf. on Signal Image Technology & Internet Based Systems (SITIS)*, 2014.

[10] Z. Guo, Q. Wang, G. Liu, J. Guo, and Y. Lu. A music retrieval system using melody and lyric. In *Proc. of IEEE Int. Conf. on Multimedia & Expo*, 2012.

[11] P. V. Kranenburg. *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*. PhD thesis, October 2010.

[12] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 1990.

[13] M. Muller. *Information Retrieval for Music and Motion, Chapter 4: Dynamic Time Warping*, pages 69–84.

[14] M. Muller, N. Jiang, and P. Grosche. A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing. *IEEE Trans. on Audio, Speech & Language Processing*, 21(3):531–543, 2013.

[15] D. Raja. *Hindustani Music: A Tradition in Transition*. D. K. Printworld, 2005.

[16] P. Rao, J. C. Ross, and K. K. Ganguli. Distinguishing raga-specific intonation of phrases with audio analysis. *Ninaad*, 26-27(1):59–68, December 2013.

[17] P. Rao, J. C. Ross, K. K. Ganguli, V. Pandit, V. Ishwar, A. Bellur, and H. A. Murthy. Classification of melodic motifs in raga music with time-series matching. *Journal of New Music Research (JNMR)*, 43(1):115–131, April 2014.

[18] S. Rao, J. Bor, W. van der Meer, and J. Harvey. *The Raga Guide: A Survey of 74 Hindustani Ragas*. Nimbus Records with Rotterdam Conservatory of Music, 1999.

[19] S. Rao and P. Rao. An overview of Hindustani music in the context of Computational Musicology. *Journal of New Music Research (JNMR)*, 43(1), April 2014.

[20] V. Rao and P. Rao. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Trans. on Audio, Speech & Language Processing*, 18(8), 2010.

[21] J. C. Ross, T. P. Vinutha, and P. Rao. Detecting melodic motifs from audio for Hindustani classical music. In *Proc. of Int. Soc. for Music Information Retrieval (ISMIR)*, October 2012.

[22] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[23] A. Srinivasamurthy, G. K. Koduri, S. Gulati, V. Ishwar, and X. Serra. Corpora for music information research in Indian art music. In *Proc. of Int. Computer Music Conf. / Sound and Music Computing Conf.*, September 2014.

[24] Y. Tanaka, K. Iwamoto, and K.Uehara. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning*, 58:269–300, 2005.

[25] Y. Tanaka and K. Uehara. Discover motifs in multi-dimensional time-series using the Principal Component Analysis and the MDL principle. In *Proc. of Int. Conf. on Machine Learning & Data Mining in Pattern Recognition*, pages 252–265, 2003.

[26] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli. Matching incomplete time-series with Dynamic Time Warping: An algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, 45(1):11–34, 2008.

[27] A. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In *Proc. of ACM Int. Conf. on Multimedia*, pages 57–66, 1999.

[28] A. Vahdatpour, N. Amini, and M. Sarrafzadeh. Towards unsupervised activity discovery using multi-dimensional motif detection in time-series. In *Proc. Of Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2009.