MELODIC CONTOUR-BASED QBH SYSTEMS: ANALYTICAL MODELING AND PERFORMANCE EVALUATION

Sumantra Dutta Roy, Preeti Rao, Ameya Shekhar Galinde and Rishabh Bhargava

Department of Electrical Engineering,
IIT Bombay, Powai, Mumbai - 400 076, INDIA.
{sumantra,prao,ameya,rishabh}@ee.iitb.ac.in

ABSTRACT

The paper proposes an analytical modeling of important parameters in a melody-based Query-by-Humming system, and proposes a new function to characterise the performance of such systems. Results of experiments with analytical models, as well as an actual QBH system give results consistent with empirical results mentioned in the literature.

1. INTRODUCTION

Most Query-by-Humming (hereafter, QBH) systems operate on the pitch contour alone [1], [2], [3], [4], [5], [6], [7], as opposed to those that operate on other features of audio input, such rhythm [8]. However, there have been few approaches to analytical modeling and performance analysis of QBH systems. Performance analysis methods primarily consist of identification of errors [9] and gross retrieval statistics (Precision and Recall) [10]. As far as analytical modeling is concerned, related work in the area has only considered the use of a 3-, 5- or 7-level pitch contour [4], [3], [6] - this is usually based on empirical studies. To the best of our knowledge, our earlier work [11] has been the only attempt to derive such an estimate using analytical methods. We extend the ideas of our earlier work to build a more complete and comprehensive analytical study of QBH systems. We develop a new coefficient to evaluate the performance of a QBH system. Our results of experiments with analytical models, as well as an actual melody database give results consistent with those in the existing literature.

1.1. Representation; Performance Measures

We assume that all notes in a musical piece lie only on a quantised set of absolute pitch values (in Hz). A common representation assumes the interval between two such adjacent values to be a 'semitone', and an ensemble of 12 such pitch values or notes is an 'octave' [2]. A logarithmic scale is convenient for this purpose, with a least count of a semitone. As in the work of Kim *et al.* [4], we deal with *relative notes* as opposed to absolute notes. This imparts invariance to *pitch transposition i.e.*, the same melody correctly sung at two different musical scales have the same representation.

Kageyama *et al.* [7] and Ghias *et al.* [5] propose the use of static heuristic thresholds for splitting the melodic contour into the desired number of levels. Sonoda *et al.* [6] propose the use of dynamic determination of thresholds for a 3-level contour. Kim *et al.* [4] consider empirical evidence to decide on choosing the number of levels of quantisation, k. Our earlier work [11] proposes a method to derive the optimal number of quantisation levels, given statistics about the melodies in the database, and sample user query statistics. One of the main limitations of this approach was its applicability to a uniform quantisation. In the current paper, we remove this and build a generic framework, applicable to any general case of quantisation. We further generalise our work to handle query strings of any length, and propose a coefficient to characterise the performance of a QBH system. We compare our results with existing measures of performance.

2. STATEMENT OF THE PROBLEM

We consider a range of relative notes R. We assume that the finest level of quantisation will give rise to N relative notes, lying between two limits r_0 and r_N, respectively[1]. *Given a $N-$ level quantisation, we wish to divide this range into k intervals (using $k-1$ markers between r_0 and r_N) - The aim is to find an optimal value for k.* Two desirable requirements governing the choice of a suitable value of k are:

- *Fidelity*: a desirable requirement is to have a close match between the hummed contour, and a close one from the database.
- *Robust Matching*: a strategy should ideally have adequate robustness to cater to different untrained singers, who may occasionally go off-key.

The task at hand is to find this optimal value of k, as described above. In the following section, we propose a new function to account for these mutually contradictory requirements, a minimum of which will result in an optimal value of k.

3. THE DEMERIT COEFFICIENT

We define the **Demerit Coefficient** $\mathcal{M}_D(k,\mu)$ for a Database of songs D, as follows:

$$\mathcal{M}_D(k,\mu) \overset{\triangle}{=} \mu\mathcal{F}_D(k) + (1-\mu)\mathcal{R}_D(k) \quad (1)$$

Here, $\mathcal{F}_D(k)$ and $\mathcal{R}_D(k)$ represent the **Fidelity** and **Robust-Match** functions, respectively, which we define in the following sections, below. (Section 3.1 and Section 3.2, respectively.) μ is an arbitrary scalar coefficient which specifies the required relative percentage of the two constituent terms in the expression for $\mathcal{M}_D(k,\mu)$. The task at hand is to find *argmin*$_k$ $\mathcal{M}_D(k,\mu)$ *i.e.*, that value of k for which $\mathcal{M}_D(k,\mu)$ achieves a minimum value.

Commonly used methods of dividing the relative note axis include having uniform quantisation, static heuristic thresholds as in the systems of Kageyama *et al.* [7] and Ghias *et al.*[5], or dynamically determined thresholds of Sonoda *et al.* [6]. The cases of uniform quantisation and that of Sonoda *et al.* are opposite in character. The advantage of the former is the relative simplicity, since it involves a smaller number of parameters. The latter seeks to divide the intervals into equal probability masses - optimal for the particular system in question. In this paper, we show results of our formulation using both the above cases - one can handle the case of static heuristically determined thresholds on similar lines. We first propose our framework for the equal probability mass formulation of Sonoda *et al.* [6]. In Section 4, we present our formulation attuned to the simpler case of uniform thresholds.

We assume the k intervals to be numbered 0 to $k-1$, with $k+1$ markers $m[i]$ appropriately placed (according to the strategy used for splitting the relative note axis) in the range r_0 to r_N. The ith interval characterised by the range of relative notes r_j: $m[i] \leq r_j < m[i+1]$. We define $p^D[x]$ as the discrete probability of a particular relative note x. This is a characteristic of a particular a characteristic of the particular database, and depends its constituent songs. Figure 1 shows samples of such curves from our database of songs, as well as an MIT database [4]. For the equal probability mass formulation of each interval, we have

$$\sum_{m[i]\leq x<m[i+1]} p^D[x] = \frac{1}{k}, \quad \forall\ i,\ 0 \leq i \leq k-1 \quad (2)$$

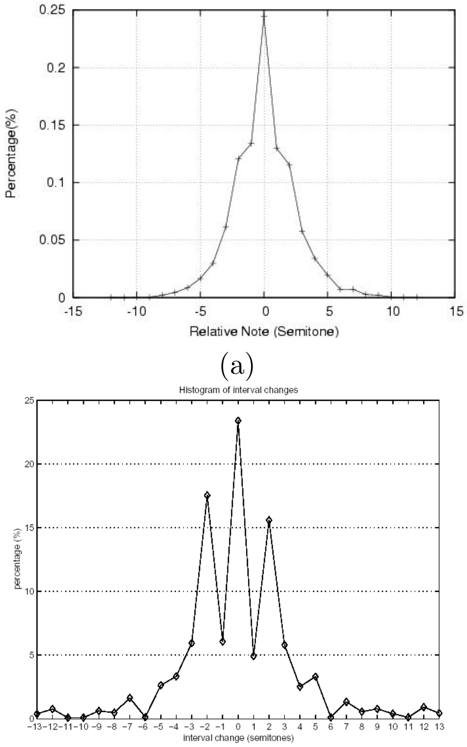
3.1. The Fidelity Term $\mathcal{F}_D(k)$

We define the Fidelity Term $\mathcal{F}_D(k)$ as follows:

$$\mathcal{F}_D(k) \overset{\triangle}{=} \frac{1}{\widehat{\mathcal{F}_D(k)}}[\sum_{\forall\ x}[x - ind[x]]^2\ p^D[x]]^{1/2} \quad (3)$$

Here, the summation is over all relative notes x in the songs in the given database D, and $p^D[x]$ denotes the discrete probability of a particular relative note x. We define the *Interval Indicator Func-*

[1] The conversion between the two discrete scales $[r_0, r_N]$ and $[r'_{-N/2}, r'_{N/2}]$ is a straightforward linear and invertible function. Throughout this paper, we use the two interchangeably.



(a)

(b)

Figure 1: Distribution of relative notes: $p^D[\cdot]$ for (a) our database, and (b) an MIT database (taken from [4])

tion $ind[x]$ for a relative note x as follows:

$$ind[x] \overset{\triangle}{=} \max_j m[j];\ \ m[j] \leq x \qquad (4)$$

In other words, $ind[x]$ indicates the left relative note which characterises an interval - if $ind[x] = m[s]$ for a given relative note x, the interval of relative notes r_j in consideration is $m[s] \leq r_j < m[s+1]$. $\widehat{\mathcal{F}_D(k)}$ is a normalising factor. We may take this as $\max_j |m[j+1] - m[j]|$, $0 \leq j \leq (k-1)$ for example, or simply the maximum of the terms being summed up.

Having k intervals implies that for a given relative note x, all relative notes r_j lying in the discrete interval $m[s] \leq r_j < m[s+1]$ would be characterised by a point, which we consider (without loss of generality) as being the left limit of the interval. Thus, $\mathcal{F}_D(k)$ is a measure of the average deviation of relative notes which would get classified by the k-level system as being in its corresponding interval.

3.2. The Robust-Match Term $\mathcal{R}_D(k)$

We define the Robust-match term $\mathcal{R}_D(k)$ as follows:

$$\mathcal{R}_D(k) \overset{\triangle}{=} \frac{1}{\widehat{\mathcal{R}_D(k)}} \sum_{\forall\ x} [\sum_y (y-x)^2\ p_x^U[y]]^{1/2}\ p^D[x] \qquad (5)$$

The outer summation is over all relative notes x in the songs in the given database D. The summation for y is over all relative notes which are not in the same interval as the relative note x *i.e.*, $\{r_0 \leq y < m[s]\} \ \cup\ \{m[s+1] \leq y \leq r_N\}$. Here, $m[s]$ is the same as that used in the previous section (Section 3.1) *i.e.*, we take $ind[x]$ as $m[s]$ for a given relative note x, so that the notes r_j in the same relative note interval as x have $m[s] \leq r_j < m[s+1]$. Here, $\widehat{\mathcal{R}_D(k)}$ is the normalisation factor. We may take this to be $R - \max_j |m[j+1] - m[j]|$ for instance, or simply the maximum of the terms being summed up. $p_x^U[\cdot]$ is the one-dimensional probability mass function of all user query relative notes for a given relative note x. (Figure 2 shows the probability mass function of $p_x^U[\cdot]$ for different relative notes x, in our database)

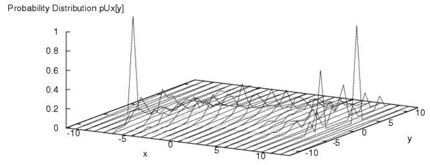


Figure 2: The probability mass function $p_x^U[y]$ for relative notes in our database (The Robust-Match Function $\mathcal{R}_D(k)$, Section 3.2)

We note that $p_x^U[\cdot]$ accounts for typical user characteristics for a given relative note x. QBH systems typically employ common techniques of Melodic contour matching such as those based on Dynamic Programming [12]. The optimal match gives us *a correspondence* between the notes of the query melodic string, and a database entry - the reference contour. We use this information to build up our $p_x^U[\cdot]$ estimates. The use of the $p_x^U[\cdot]$

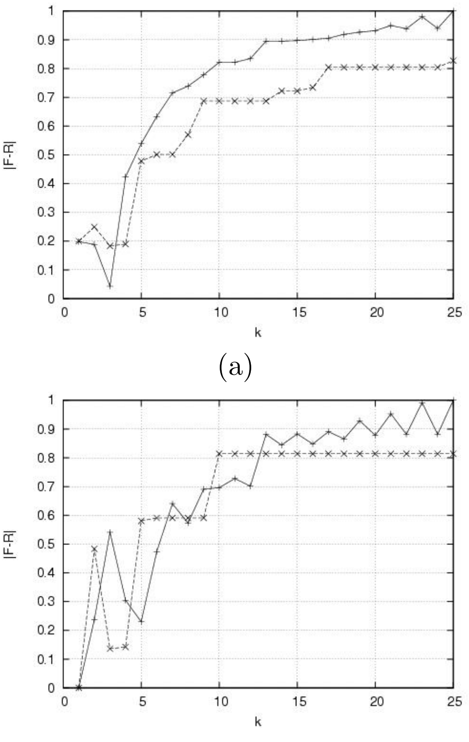
function also subsumes the concept of melodic similarity based on chords [13].

3.3. Finding the Optimal k

We differentiate Equation 1 with respect to the two variables k and μ, and set these to zero. We can find out the value of k from the partial differentiation with respect to μ. We evaluate $|\mathcal{F}_D(k) - \mathcal{R}_D(k)|$ for varying values of k, and check for minima close to zero. Further, we can find the optimal value of μ by numerical differentiation. We evaluate $|\frac{\delta \mathcal{M}_D(k,\mu)}{\delta k}|$ for $\delta k = 1$ (the smallest possible discrete change in k), and find out the value of μ, for which $|\delta \mathcal{M}_D(k, \mu)|$ is minimum. This is the required value of μ.

We have experimented with statistics from an existing QBH system TANSEN [3], as well as with representative models. Our database has 201 song phrases, with an average of 26.52 notes in each. We have built up the $p_x^U[\cdot]$ statistics from 936 user queries. The solid curve in Figure 3(a) shows a plot of $|\mathcal{F}_D(k) - \mathcal{R}_D(k)|$ for varying values of k. The optimum corresponds to $k = 3$ and $\mu = 0.01$. The portions of flatness in the curve are due to the nature of the floor function in l_k and hence, $|\mathcal{F}_D(k) - \mathcal{R}_D(k)|$ as well. For our experimentation, we have considered both the normalisation factors $\widehat{\mathcal{F}_D(k)}$ and $\widehat{\mathcal{R}_D(k)}$ as the maximum term in the respective summations.

A set of analytical models permits one to experiment by changing various system parameters. From Figures 1 and 2, we can approximate the $p^D[\cdot]$ curve by a suitable Gaussian, and $p_x^U[\cdot]$ as a Gaussian with standard deviation proportional to x. Figure 3(b) shows the corresponding plot, which gives the optimal values of k and μ as 5 and 0.01. The solid curve in Figure 4 shows the variation in the optimal value of k with the variance of the $p^D[\cdot]$ Gaussian for this case (equal probability mass).



(a)

(b)

Figure 3: The optimal k for the equal probability mass case (solid curve) and the uniform quantisation case (broken curve): $|\mathcal{F}_D(k) - \mathcal{R}_D(k)|$ vs. k for (a) actual database statistics; and (b) analytical simulation: $p^D[\cdot]$ and $p_x^U[\cdot]$ as Gaussians

4. UNIFORM QUANTISATION OF THE RELATIVE NOTE AXIS

For the case of uniform quantisation, we define l_k as the 'length' of an interval along the relative note axis, as follows: $l_k \triangleq \lfloor (r_N - r_0)/k \rfloor$, where the $\lfloor\ \rfloor$ notation denotes the largest integer smaller than the number. For the uniform quantisation case, we define the Fidelity Term $\mathcal{F}_D(k)$ as follows:

$$\mathcal{F}_D(k) \triangleq \frac{1}{\widehat{\mathcal{F}_D(k)}}[\sum_{\forall\ x}[x - l_k(x\ div\ l_k)]^2\ p^D[x]]^{1/2} \qquad (6)$$

We may take the normalising factor $\widehat{\mathcal{F}_D(k)}$ as $l_k - 1$ for example, or simply the maximum of the terms being summed up. The definition of the Robust-Match term remains unchanged (Equation 5) in the uniform quantisation case. Only the definition of the intervals is different, here. The inner summation (for y) is for all relative notes which are not in the same interval as the relative note x *i.e.*, $\{r_0 \leq y < l_k(x\ div\ l_k)\} \cup \{l_k(x\ div\ l_k + 1) \leq y \leq r_N\}$. In this case, we may the normalising factor $R - l_k$ $(= r_N - r_0 - l_k)$, or simply the max-

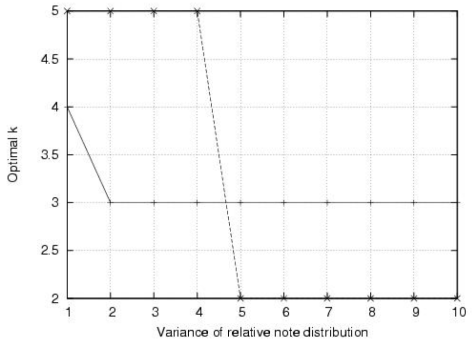


Figure 4: Variation of the optimal k for different σ values: simulation, with $p^D[\cdot]$ assumed to be a Gaussian. Solid curve shows the equal probability mass case, and the broken one, the uniform quantisation case.

imum of the terms being summed up. Our previous work [11] deals with the uniform quantisation case in detail. In Figures 3(a) and (b), and 4, the broken curve shows the uniform quantisation case. The optimal values of k and μ for this case for actual data and simulated data are 3 and 0.01 for both cases, respectively.

REFERENCES

[1] C. Meek and W. Birmingham, “Johnny Can’t Sing: A Comprehensive Error Model for Sung Music Queries,” in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2002.

[2] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Sunningham, “Toward the Digital Music Library: Tune Retrieval from Acoustic Input,” in *Proc. ACM Digital Libraries*, 1996.

[3] M. Anand Raju, B. Sundaram, and P. Rao, “TANSEN: A Query-By-Humming based Music Retrieval System,” in *Proc. National Conference on Communications (NCC)*, 2003.

[4] Y. E. Kim, W. Chai, R. Garcia, and B. Vercoe, “Analysis of a Contour-based Representation for Melody,” in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, October 2000.

[5] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, “Query By Humming - Musical Information Retrieval in an Audio Database,” in *Proc. ACM Multimedia*, 1995.

[6] T. Sonoda, M. Goto, and Y. Muraoka, “A WWW-based Melody Retrieval System,” in *Proc. ICMC*, October 1998, pp. 349 – 352.

[7] T. Kageyama, K. Mochiezuki, and Y. Takashima, “Melody Retrieval with Humming,” in *Proc. ICMC*, 1993, pp. 349 – 351.

[8] J. C. C. Chen and A. L. P. Chen, “Query by Rhythm: An Approach for Song Retrieval in Music Databases,” in *Proc. Workshop on Research Issues in Database Engineering*, 2003, pp. 122 – 128.

[9] C. Meek and W. Birmingham, “The Dangers of Parsimony in Query-by-Humming Applicaitons,” in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2003.

[10] J. L. Hsu, A. L. P. Chen, H. C. Chen, and N. H. Liu, “The Effectiveness Study of Various Music Information Retrieval Approaches,” in *(CIKM)*, November 2002.

[11] S. Dutta Roy, P. Rao, and A. S. Galinde, “Contour-Based Melody Representation: An Analytical Study,” in *Proc. National Conference on Communications (NCC)*, 2004, pp. 536 – 540.

[12] L. Prechelt and R. Typke, “An Interface for Melody Input,” *ACM Transactions on Computer-Human Interaction*, vol. 8, no. 2, pp. 133 – 149, 2001.

[13] T. C. Chou, A. L. P. Chen, and C. C. Liu, “Music Databases: Indexing Techniques and Implementation,” in *International Workshop on Multi-Media Database Management Systems*, 1996, pp. 46 – 53.