# NEW QUERY-BY-HUMMING MUSIC RETRIEVAL SYSTEM
# CONCEPTION AND EVALUATION
# BASED ON A QUERY NATURE STUDY

*Matthieu Carré - Pierrick Philippe - Christophe Apélian*

France Telecom R&D - DIH/HDM
4 rue du Clos Courtel - BP91226
35510 Cesson Sevigne Cedex, France
firstname.lastname@rd.francetelecom.com

## ABSTRACT

In this article, we propose a new Query-by-humming Music Retrieval System, focusing on the nature of hummed queries, and more precisely, on their non-tempered characteristics. We show that avoiding pitch query quantization gives better retrieval results. The study of frequential imprecision of hummed melodies also allows us to present a new and easy way of stimulating systems for their quality evaluation.

## 1. INTRODUCTION

Musical information access is a crucial stake regarding the huge quantity available, and worldwide interest. Classical means of indexing (textual annotation) is insufficient for efficient retrieval. Usual description (title, author...) is far-removed from audio content, and needs important human intervention.

The new ISO/MPEG-7 standard, formally called Multimedia Content Description Interface, deals with (semi-)automatic descriptions of the real content of documents. Concerning music, MPEG-7 standardizes melody descriptions, especially for Query-by-Humming Music Retrieval Systems (QbHMRS). MPEG-7 doesn't normalize ways of using the descriptions (e.g. similarity measures for the comparison of descriptors) [1].

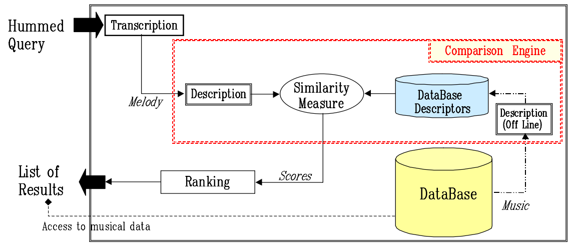

Figure 1: *Scheme of a Query-by-Humming Music Retrieval System.*

In this article, we present a new melodic comparison engine, i.e. melody descriptors and a similarity measure (cf. figure 1). Focusing on hummed query characteristics, we will distinguish them from database melodies. Our melodic descriptors will take into account the differences in precision noticed between them. Moreover, the knowledge about hummed melodies will allow us to synthesize artificial queries, providing an easy and realistic way of stimulating systems for their quality evaluation. Comparing several frequential based comparison engines, we show the superiority of the non quantized pitch query approach, for Query-by-Humming Music Retrieval.

## 2. PREVIOUS WORK

Previous work in melody retrieval by humming has mainly focused on the comparison engine. The question of the nature of database involved is avoided for the moment because of the lack of efficiency in the transcription system. Extracting the musical score from any polyphonic music is not yet possible, so systems generally use MIDI type databases. This makes score available. Concerning the melodic description, the query is usually considered in the same way as the database melodies. This somehow may be true for the piano queries, for example, but it's certainly not for the hummed ones, whose pitch values are non-tempered. Thus, using database-type melodies as queries, makes system testing unrealistic.
The general trend is to represent the melodies as sequences of states (of variations : pitch and/or duration). If different states are symbolized by different symbols, the melodies can be represented by strings. Thus, they can be compared with well known string matching methods. Various work aimed to find the way to efficiently retrieve melodies from hummed queries has started from this point.

Primary work used a compact description of the melodies, freely inspired by psychologic work on memory for melodies [2]. Melody descriptors consisted in keeping only the sign of pitch variations. Three symbols $\{U,D,S\}$ were used to represent ascending variations (Up), descending ones (Down), and constancy (Same). Those systems dealt with small databases (a few hundred melodies), but with database size increase, the description had to be more precise (thus less compact) in order to ensure better discrimination [3]. At this state of maturity, the question of effective evaluation of systems' quality is raised [4]. With it, the lack of realistic system stimulation reveals the little care given until now to the proper nature of sung melodies. In our opinion, this knowledge should be taken into account when defining melodic descriptors, and also when searching for efficient ways of testing systems.

## 3. A STUDY OF HUMMED MELODIES IMPRECISIONS

The notes of hummed melodies have special properties that make them different from those of database melodies. In particular, the latters' pitch values are quantized, whereas those of hummed melodies are not. So, there are ambiguities about the pitch of sung notes performed *a cappella*.

### 3.1. Previous experiments

Only two publications have investigated the way people were singing melodies. McNab, in [3], makes people sing well-known melodies. Concerning frequential precision, he noticed that subjects tend to compress big intervals (especially those whose magnitude goes from 7 to 9 semitones), and also to extend small ones, when they belong to ascending or descending sequences. Lindsay, in [5], makes his subjects repeat the unknown melodies he plays. This allows to collect an homogeneous corpus (i.e. all intervals are equitably represented), which is not the case for McNab. Lindsay noticed that the subjects' inaccuracy could be considered as independent of the magnitude of the intervals they were targeting. The drawback of Lindsay's experience is to stimulate subjects' short-term memory, setting out of a realistic framework of QbHMRS use.

### 3.2. Experimental strategy and observations

We made our 9 subjects sing a great amount of well-known melodies. Thus, our 500 melodies corpus (more than 5 times bigger than McNab's one) allows us to compare our results to both experiments. We noticed that small intervals (1 to 4 semitones) were generally compressed, and those with a magnitude of 5 semitones were generally extended. Bigger intervals didn't present a real trend, maybe because of the smaller amount of data available. In our most represented intervals (0 to 5 semitones), the *variation* of accuracy is similar to the one revealed by Lindsay. However, the error's *magnitude* is lower than in Lindsay's observations. Although this could come from corpus differences (subjects, melodies...), this could also illustrate the best precision of long-term memory. We also noticed that, within the first four intervals of hummed melodies, error could be higher than the mean value (especially in the first and the third ones).

Although our corpus is quite large, we are still limited by the fact that the intervals considered are not represented equitably. As it seems impossible to find well-known melodies which would avoid this drawback, further studies should try to collect the largest corpus possible (subjects, intervals, and also melodic contexts...).

### 3.3. Conclusions and Modeling

Considering the variations noticed[1] were too small to be taken in account, we modeled the inaccuracy of hummed melodies, merging the 5345 interval errors available. The repartition of these errors is shown in figure 2, associated with the generalized gaussian model presented in expression 1.

$$G_g(x) = 1.06\ e^{-|1.98*x|^{1.23}} \tag{1}$$

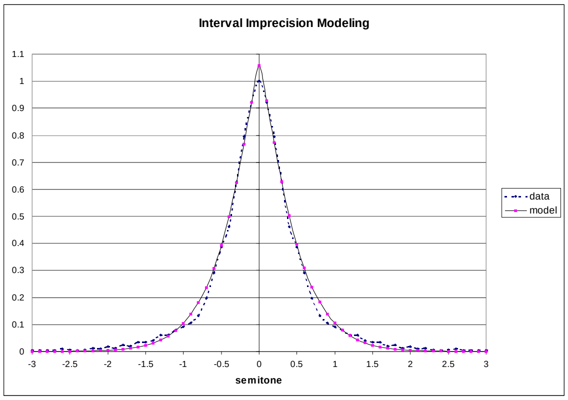

Figure 2: *Interval errors repartition and its modeling.*

More than 25% of interval errors are over a quartertone magnitude (threshold of note ambiguity). This shows it's worth taking this imprecision into account when enabling hummed queries for Music Retrieval. The model we've just presented will allow us, in section 6, to create some artificial hummed queries, facilitating system testing.

## 4. HIGH PRECISION FOR MELODIC DESCRIPTION

What we present here is a new way of considering melodic material description for Music Retrieval by Humming, distinguishing the database melodies from the hummed queries. Maximum precision of representation is provided for both melody types (database ones and hummed ones). In this paper, we focus on the discrimination properties of melody frequential information. Thus, the melody descriptors introduced here have no temporal information. Furthermore, note insertions/omissions are not treated here. As pitch imprecision is a *permanent* phenomenon, our first work investigates it exclusively.

As the database melodies are already quantized (Midi coding), there's no ambiguity about the pitch of the notes played. A precision greater than a semitone is thus not required. The hummed query case is different. To be closer to material given by the user, we will conserve the maximum precision of frequential information. The query description will be based on *non quantized* pitch values.

In systems using query quantization, very small variations (a hundredth of a semitone) of sung pitch values can lead to big changes in similarity distance. This phenomenon occurs in particular melodic contexts that don't justify such consequences. So, this makes the list of results both lose its discrimination properties and, in some way, ”randomly” disordered (as the melodies aren't treated equitably). Refusing pitch query quantification makes the scores respect the query variation proportions, providing better retrieval results.

## 5. PITCH-BASED COMPARISON ENGINE

The first type of comparison engine we investigate in this paper uses pitch sequences descriptors, which we call *pitch profiles*. A

[1] variations following the magnitude, and also the rank of intervals in the melody.

second comparison engine, with descriptors based on intervals sequences, will be considered in section 7.

### 5.1. Descriptors

The database melodies being, by nature, quantized, their descriptors are vectors of semitone *quantized* successive pitch values. The query descriptor is a vector of *non-quantized* successive pitch values. Let $\bar{q} = [q_0, \ldots, q_{N-1}]$, the descriptor of a hummed query of $N$ notes ; and $\bar{d} = [d_0, \ldots, d_{N-1}]$, the descriptor of a melodic portion of the database.

### 5.2. Similarity Measure

The similarity measure between the query and a melodic portion of the database is given by the distance between their descriptors. The score of a document will be the smallest distance found, searching through all the melodies it contains.

These descriptors are *not* free from tonality, so they have to be adjusted before the distance is computed. As tonality extraction gives ambiguous results when starting from few notes (p. 80, in [6]), an offset $\lambda$ is used to minimize the distance computed. So, in the expression of similarity measure presented below, a *mathematical criterium* (minimization of a distance) is used to overcome the ignorance of a *musical notion* (tonality).

$$D_{p\gamma}(\bar{q}, \bar{d}) = \sum_{j=0}^{N-1} \mid q_j - d_j - \lambda \mid^{\gamma} \qquad (2)$$

We considered only the two cases $\gamma = 1$, and $\gamma = 2$. For the first one, $\lambda = median(\bar{r} - \bar{d})$, and for the second one, $\lambda = mean(\bar{r} - \bar{d})$. These giving very close results, we will only present the $\gamma = 2$ case (which furthermore allows faster computing).

Example : *Let's consider a hummed query and the melodic portion it targets (it consists in the first notes of Beethoven's fifth symphony). Their pitch information is given table 1. The third and the fifth line (with "# midi") contain the two pitch profile descriptors to compare. With those values, the adjustment $\lambda$, which is the mean of the differences ($\gamma = 2$), is equal to 3.5125. So the distance between the hummed query and the melodic portion it targets is equal to* $\sum_{j=0}^{7}(Query(j) - Target(j) - 3.5125)^2$ *= 0.24875.*

| Rank | j=0 | j=1 | j=2 | j=3 | j=4 | j=5 | j=6 | j=7 |
|---|---|---|---|---|---|---|---|---|
| Query (Hz) | 166.5 | 166.2 | 166.3 | 131.3 | 144.8 | 146.0 | 146.2 | 121.7 |
| Query (# midi) | 39.7 | 39.6 | 39.7 | 35.6 | 37.3 | 37.4 | 37.4 | 34.2 |
| Target (notes) | G3 | G3 | G3 | E ♭3 | F3 | F3 | F3 | D3 |
| Target (# midi) | 43 | 43 | 43 | 39 | 41 | 41 | 41 | 38 |

Table 1: *Pitch values for distance computing example.*

## 6. QUALITY EVALUATION OF SYSTEMS

To evaluate the retrieval quality of QbHMRS, we use a recall criteria : the number of relevant documents retrieved divided by the total number of relevant documents [7].

The relevant documents are defined in the following way : The melody targeted by the user is manually extracted from the database, then injected in any of the systems listed in this paper, excepted UDS. As the melody targeted constitutes a perfect sung query, the configurations numbered by 1, 2, 4, 5, and 6 would give the same result. Within the list of responses (limited to the 15 best matches in our systems), the ones whose score is 0 are considered as references (perfect matches). Comparing results of natural queries (the 500 melodies of section 3) to those references gives recall performance of the system tested.

### 6.1. Database

Our database contains about 20,000 midi files. All tracks (average of 6.7 tracks per file) can be targeted, excepted drum tracks whose events doesn't correspond to melodic information. Polyphonic tracks are transformed into monophonic melodies, following reduction rules defined by Uitdenbogerd [8]. Representing more than 37,000,000 indexed notes, this is, to our knowledge, the biggest database used until now.

### 6.2. Tests

The first three system configurations tested are the following :

1. *NonQuant_PP* : The query descriptor consists in a Non Quantized Pitch Profile. The similarity measure is the one we've just presented in section 5 (expression 2 with $\gamma = 2$);
2. *Quant_PP* : The query descriptor consists in a Quantized Pitch Profile. Description is combined with the same Pitch profile distance as 1. The query's quantization is done in three steps. First, intervals are extracted from successive pitch values. Then, they're rounded to the nearest semitone value. Finally, starting from those quantized intervals, a quantized pitch profile is built. This quantization process changes the original tonality, but this has no effect because the similarity measure uses adjusted pitch profile descriptors ($\lambda$ in expression 2) ;
3. *UDS* : the pitch intervals are converted into three states, Up-Down-Same ; a distance based on string matching is used to compute the score [2]. The latter is the amount of symbol differences between the two melodic descriptors.

   *UDS* doesn't represent the state of the art in terms of QbHMRS, but as it's well-known, it's a good common basis for the comparison of systems.

Figure 3 illustrates the retrieval performances of the tested systems. We can see the good results of pitch profile based comparison engines, and the improvement gained by avoiding pitch query quantization.

### 6.3. Artificial stimuli

Testing systems with real hummed queries is a very laborious task. Collecting queries, finding the melodies they target, and defining references for recall criteria takes a lot of time. Furthermore, it's hard to collect a homogeneous corpus (users' queries have various targets and lengths). To facilitate the system testing, we propose a new way of stimulation, which is based on the error model illustrated figure 2. Starting from database extracted melodic fragments, artificial hummed requests (of any length) are synthesized. Thus, systems can be tested in a more flexible way than with real queries, and in a more realistic way than with perfect queries. Figure 4 shows recall performances (of configurations 1 to 3) estimated in this way.

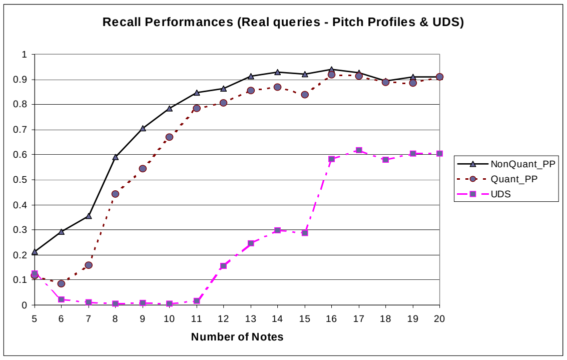

Figure 3: *Recall performances of three melodic comparison engines stimulated by real queries.*

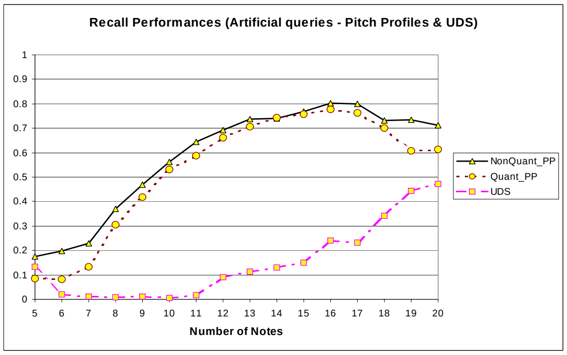

Figure 4: *Recall performances of three melodic comparison engines stimulated by artificial queries.*

Figures 3 and 4 show that, in spite of under-estimated performances, artificial queries allow the same configuration ranking as that obtained with real queries. So, our imprecision model provides guidelines, which can be trusted for systems conception, avoiding the hard preprocess due to real query testing.

In the following section, we will present another type of comparison engine. Tests with both real and artificial queries will be done.

## 7. INTERVAL-BASED COMPARISON ENGINE

The second comparison engine type considered is based on interval sequences. Using previously introduced notations, the descriptors used for query and melodic portion are respectively $[q_1 - q_0, \ldots, q_{N-1} - q_{N-2}]$, and $[d_1 - d_0, \ldots, d_{N-1} - d_{N-2}]$. Their length is $N - 1$.

As these descriptors are free from tonality, the distance can be applied straight away (no adjustment needed). The similarity measure between the query and a melodic portion of the database is then given by the expression :

$$D_{i\gamma}(\bar{q}, \bar{d}) = \sum_{j=0}^{N-2} |\, q_{j+1} - q_j - (d_{j+1} - d_j)|^{\gamma} \qquad (3)$$

with $\gamma = \{1, 2\}$. As in section 5, the two cases $\gamma = 1$, and $\gamma = 2$ giving very close results, we will only present the $\gamma = 2$ case.

Example : *Starting from the pitch information given in table 1, we obtain the interval based descriptors presented in table 2. The distance between them is equal to* $\sum_{j=0}^{6}(Query'(j) - Target'(j))^2$ *= 0.17.*

| Rank | j=0 | j=1 | j=2 | j=3 | j=4 | j=5 | j=6 |
|---|---|---|---|---|---|---|---|
| Query’ ($\Delta$(# midi)) | -0.1 | 0.1 | -4.1 | 1.7 | 0.1 | 0 | -3.2 |
| Target’ ($\Delta$(# midi)) | 0 | 0 | -4 | 2 | 0 | 0 | -3 |

Table 2: *Interval based descriptors values for distance computing example.*

The system configurations tested are the following :

4. *NonQuant_IS* : The query descriptor consists in a Non Quantized Interval Sequence. The similarity measure is the distance we’ve just presented (expression 3 with $\gamma = 2$);
5. *Quant_IS* : It’s the same configuration as 4, but with a Quantized Interval Sequence for the query descriptor ;
6. *Quant_StrMat* : As in configuration 5, the query descriptor is quantized. As in configuration 3, the similarity measure is based on a String Matching technique (score = the amount of symbol differences between two melodic descriptors).

   Like the *UDS* configuration, the description uses sequences of states. However, having a finer precision, configuration 6 provides a better discrimination than that of the three states *UDS* configuration.

Recall performances (for a real queries stimulation) are presented in figure 5. For this comparison engine type too, quantization leads to worse performances. However, interval based systems seem less sensitive to it, as the degradation is smaller than that of pitch profile based systems.

Now that we have seen that pitch quantization has a negative effect for both comparison engine types, let’s stimulate interval based systems with our artificial hummed queries. Figure 6 illustrates the recall performances obtained in this way. As we can see, our artificial queries lead to a very good estimation of recall results. Providing for a right *ranking*, and also for almost right *recall values*, our imprecision model can be used with interval based systems too.

The estimation of recall performances, obtained using our artificial *stimuli*, has a different quality for the two *types* of comparison engines presented. Assuming interval errors independence, our model is best suited to interval based systems, whose distance is made up of *local* differences, whereas pitch profile based systems make a more *global* calculation (adjustment $\lambda$ depends on the whole values of the descriptors). This shows that our imprecision modeling, although giving satisfaction, would be improved by taking into account the interval errors’ dependence. Thus, comparison engines of different types could be compared equitably within a single test.

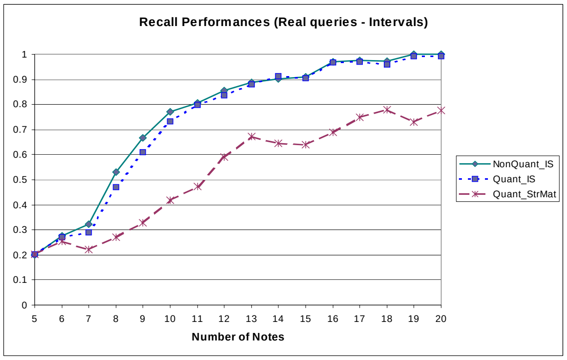

Figure 5: *Recall performances of three interval based melodic comparison engines stimulated by real queries.*

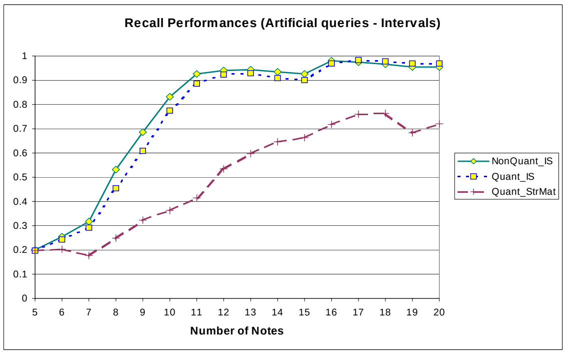

Figure 6: *Recall performances of three interval based melodic comparison engines, stimulated by artificial queries.*

## 8. BEST COMPARISON ENGINE

To conclude on the best comparison engine presented, let's compare recall performances (using real queries) of four of the configurations already seen :

- Configuration 1 i.e. *NonQuant_PP*
- Configuration 4 i.e. *NonQuant_IS*
- Configuration 5 i.e. *Quant_StrMat*
- Configuration 3 i.e. *UDS*

Figure 7 shows there is no absolute winner. *NonQuant_PP* gives best results for queries from 5 to 15 notes, then *NonQuant_IS* takes the advantage. As our collected queries have an average length of 13 notes[2], we consider the Non Quantized Pitch Profile based configuration, as the best comparison engine.

Our non quantized approach gives very good results in a error context limited to frequential imprecisions. Further work will consist in taking into account note insertions/omissions. This could be based on the matching of small overlapped parts of actual descriptors, or considering temporal information of melodies.

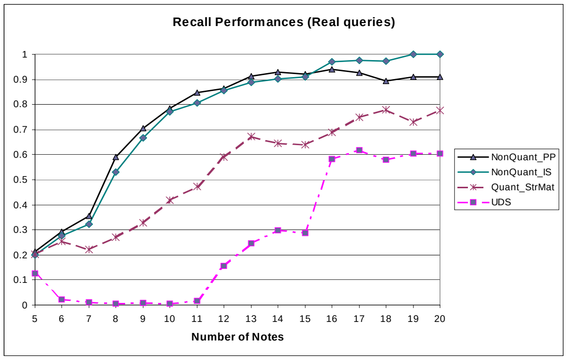

Figure 7: *Recall performances of four melodic comparison engines, stimulated by real queries.*

## 9. CONCLUSION

In this article, we have shown that studying the hummed queries' nature allowed us to provide a new efficient Query-by-humming Music Retrieval System. We showed that avoiding pitch query quantization leads to better retrieval performances. Modeling pitch query imprecisions also allowed us to synthesize artificial hummed queries. Avoiding the laborious collection and analysis of real hummed queries, they provide an easy and realistic way of stimulating systems for their quality evaluation.

## 10. REFERENCES

[1] http://www.mpeg-7.com/

[2] Ghias, A., Logan, J., Chamberlin, D., and Smith, B.C., ”Query By Humming, Musical Information Retrieval in an Audio Database,” Proc. of ACM Multimedia Conf., pp. 231-236, 1995.

[3] McNab, R.J., Smith, L.A., Brainbridge, D., and Witten, I.H., ”Tune Retrieval in the Multimedia Library,” Multimedia Tools and Applications, 10, pp. 113-132, 2000.

[4] Downie, J.S., ”Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams a text,” Ph.D. thesis, The University of Western Ontario, London, Ontario, 1999.

[5] Lindsay, A., ”Using Contour as a Mid-level Representation of Melody,” Report of Master of Science in Media Arts and Sciences, 1996.

[6] Krumhansl, C.L., ”Cognitive Foundations of Musical Pitch,” New York Oxford University Press, 1990.

[7] Salton, G., and McGill, M.J., ”Introduction to Modern Information Retrieval,” McGraw-Hill, New York, 1983.

[8] Uitdenbogerd, A.L., and Zobel, J., ”Melodic Matching Techniques for Large Music Databases,” Proc. of ACM Multimedia Conf., pp. 57-66, 1999.
http://www.kom.e-technik.tu-darmstadt.de/acmmm99/ep/uitdenbogerd/

[2] Only melodies whose query length is free were considered for the mean value presented. That's why it can't be deduced from the amount of queries and the amount of interval errors cited in section 3.