# The MAMI Query-By-Voice Experiment: Collecting and annotating vocal queries for music information retrieval

**Micheline Lesaffre[1], Koen Tanghe[1], Gaëtan Martens[2], Dirk Moelants[1], Marc Leman[1], Bernard De Baets[3], Hans De Meyer[2] and Jean- Pierre Martens[4]**

[1] IPEM: Department of Musicology, Ghent University, Blandijnberg 2, 9000-Ghent, Belgium
Micheline.Lesaffre@UGent.be
[2] Department of Applied Mathematics and Computer Science, Ghent University
[3] Department of Applied mathematics, Biometrics and Process Control, Ghent University
[4] Department of Electronics and Information Systems (ELIS), Ghent University

## Abstract

The MIR research community requires coordinated strategies in dealing with databases for system development and experimentation. Manually annotated files can accelerate the development of accurate analysis tools for music information retrieval. This paper presents background information on an annotated database of vocal queries that is freely available on the Internet. First we outline the design and set up of the experiment through which the vocal queries were generated. Then attention is drawn to the manual annotation of the vocal queries.

## 1 Introduction

Query by vocal input is a paradigm for the specification of musical audio content in the context of music information retrieval (MIR). Such systems typically make a distinction between the query part (specification of musical content by users) and the target part (database of audio files). The MAMI project (http://www.ipem.UGent.be/MAMI/) aims at building a music retrieval system using advanced audio-mining techniques (Leman et al., 2002). Paying attention to user-friendly interaction with the query part is a prerequisite for future successful applications of audio-based MIR. Little is known about the capabilities and behavior of users dealing with audio-based MIR, such as the impact of memory recall, human inaccuracy in the performance of vocal queries and distinction between different user profiles. In context of this research a query-by-voice experiment was set up that generated a query file database of around 1500 queries. By 'query-by voice' we mean queries produced by the voice and the vocal organs, notably singing lyrics, singing syllables, humming or whistling. This query file database was manually annotated in view of analysis of spontaneous user behavior and evaluation of tool effectiveness.



A database of manually annotated vocal queries is useful for multiple reasons. The major motivations are that it leads to a better understanding of the user's querying behavior, it is a reference for automated annotation development and it serves as a benchmark for testing MIR systems. This paper introduces the MAMI (Musical Audio Mining) query-by-voice experiment. The aim of the experiment was to collect vocal queries for detailed study of spontaneous user behavior and for setting up a database that can be used for developing and testing query-by-voice based MIR systems. For this experiment, 30 pieces of music were selected, from a larger MAMI target database. The selection contains popular music, ranging from chanson to heavy metal, well-known Flemish children songs and classical music. Thus it reflects the heterogeneous musical landscape of today (for a detailed list see the MAMI website). Seventy-two subjects were involved in the experiment. We start with a brief description of how the experiment was set up. Then the methods used for manual annotation are discussed and some results of the statistical analysis are presented.

## 2 Experiment setup

This section contains a brief description of the experiment. A use case description gives an overview of the course of the experiment. It is followed by a summary of the input and output files and some notes about the software itself.

### 2.1 Use case description

#### *2.1.1 Physical setup*

A PC running Windows98SE is installed on a table in a small, closed room with no special sound isolation (environmental noise and sounds are still slightly hearable). This is a compromise between recording in a natural environment and giving the user some feeling of privacy. A low-budget microphone typically shipped with a new multimedia PC (in our case a Labtec Verse 514) is connected to the microphone

input of the computer's sound card (a standard sound card, in our case a Yamaha DS1 PCI). Headphones are connected to the sound card output. The subject is sitting in a chair in front of the computer monitor and the microphone, which is fixed on top of the monitor keeping a distance of about 30 centimeters of the subject's head. The experiment takes about 35 minutes.

#### *2.1.2 Sound level check*

The program can be told to perform a sound level check first. This should be done at least once to check whether the recording settings of the microphone/sound card are (still) OK. The sound card settings should be adjusted so that the level is "just high enough without causing clipping for loud singing or clapping".

#### *2.1.3 Collecting info on the subject*

Each subject is automatically given a unique ID (a counter increased by 1 each time the program is run). This unique ID is used for labeling the generated information files. Then a basic profile of the user is obtained by asking the following questions: "What is your age?" "Are you male or female?" "How many hours a week do you actively listen to music?" "Do you play a musical instrument (yes/no)? If so, how many hours a week?" "What is the highest level of musical education you've had". This information is written to a profile file, together with the name of this file and the name of the file that is used as a log of the experiment for this subject.

#### *2.1.4 Collecting info on the subject's knowledge of the musical pieces*

This phase of the experiment collects information about the subject's knowledge of the various pieces used in the experiment. All pieces are specified in the configuration file and gathered in a set called Set1. We used 30 pieces for Set1 and for each subject this same set of 30 pieces is reorganized at random. The 30 pieces were selected from the MAMI target database that contains 160 entire pieces of music stored in WAV format. Starting with the first piece of Set1, the title is shown together with (between brackets) its composer, performer or other brief specification and the subject is asked whether he/she would be able to imitate a fragment of that piece. If the answer is "yes", the piece is added to a set of "known, imitable" pieces (Set3). If the answer is "no", the subject is asked why not. He/she can then choose between: "I don't know it" (piece is added to Set4K), "(I think) I know it, but I just can't remember how it sounds" (piece is added to Set4R) or "I really do know it, but I can't imitate it" (piece is added to Set6). Then the second piece is presented etc… This iteration over the pieces in Set1 stops if all pieces are handled, or as soon as Set3, Set4R and Set4K all contain enough pieces. The results of this categorization are also written to the log file where each of the above 4 sets shows the ID's of the pieces that were added to it.

In order to obtain enough diversity in the subject's knowledge about the pieces, we chose to use 30 pieces for Set1 and aim to get 10 pieces for Set3 (the final number depends on the subject's knowledge of the pieces). This is a tradeoff between getting enough recordings of the same piece by different subjects and getting recordings for enough different pieces. To reduce the total time of the experiment, we chose to use only 2 pieces from Set4R and 2 from Set4K (see *Experiment part 2*). Also note that there is no Set2 (for historical reasons). An overview of the different sets of pieces is given in Table 1.

| | |
|---|---|
| **Set1** | fixed set of pieces from MAMI target database |
| **Set3** | known and imitable |
| **Set4K** | not known |
| **Set4R** | thought to be known, but not remembered |
| **Set5** | fixed fragment to be imitated in different ways |
| **Set6** | known, but not imitable |

Table 1: Overview of the different sets of musical pieces

#### *2.1.5 Experiment part 1*

This part is focussed on the reproduction of known pieces from long-term memory. It gives information on how people prefer to (or would like to) imitate musical pieces and which parts are imitated without having heard the piece.

The pieces that ended up in Set3 ("known pieces") are presented one by one as described above (only title + composer, performer or other brief specification, no sound). The subject is asked to imitate the piece vocally. The following methods are proposed (but not imposed): humming, singing the text, singing using a syllable, whistling, or any mix of these methods. The subject is free to choose the fragment or voice/instrument he/she wishes to imitate. Subjects can choose when to start and stop the recording by pressing the enter key, the maximum duration is 30 seconds. After the recording, the subject is given the choice to make a second recording. This can be done when the subject is not satisfied with the first recording, if he/she wants to use another method, or wants to perform another fragment from the same piece. After the vocal imitation(s), the subject is asked whether he/she wants to describe the piece in another way. The choices are the following: make a recording (using a method other than the previous ones), provide a textual description of the piece or describe an alternative query method using typed text. Each of these choices can be made at most once.

The iteration over the pieces in Set3 stops if a predefined number of pieces (we used 10) were handled (or if all pieces were handled if there are not enough pieces in Set3 to reach this number). All recordings are stored in WAV files (44.1 kHz, 16-bit mono) and the textual inputs are stored in the log file. A list of the pieces that were presented, the names of the files that were recorded for each piece and the choices made by the user are also stored in the log file. Remark: If the subject did not specify any piece in the first part as "known, imitable" (Set3 is empty), this entire part has to be skipped and the experiment immediately continues with part 2. However, in our experiment, a number of pieces that were supposed to be familiar to a large majority of the participants had been included. This almost guarantees Set3 to contain a sufficient number of queries to obtain relevant results.

#### 2.1.6 Experiment part 2

This part is focussed on imitations from short-term memory, after listening to the piece. It is used to investigate differences with imitations from long term memory, and to get an idea of which parts of a piece tend to "stick" after just hearing the entire piece.

The subject has to listen to a number of entire pieces (we used 4), one by one. Immediately after listening to a piece he/she is asked whether it sounds familiar or not and then to imitate a fragment from it by vocal query. Again, the subject has the option to make a second recording if he/she wants to. The pieces in this experiment are selected as follows. We start with the pieces from Set4K ("not known") until we have had a predefined number of them (we used 2) or until we run out of them. Then the pieces of Set4R ("known, but not remembered") are used, again until we had a predefined number of them (we used 2) or until we run out of them. If we still have not presented enough entire pieces, the remaining pieces are selected from (in this order): Set3, Set4K, Set4R and Set6. Again, recordings are written to WAV files and all used pieces and the responses are logged in the log file.

#### 2.1.7 Experiment part 3

This last part is meant to give some information on differences in performances of the same melody by various subjects using different query methods (male/female differences, pitch fluctuations, use of vibrato, accuracy of imitation, number of people that can whistle a melody…).

A short musical fragment (Set5) is played back (same fragment for all subjects). The subject can listen to it up to three times (the first time he/she is asked whether he/she knows it or not). Then the subject is asked to imitate it in 4 different ways: sing it with words (the text is shown on the screen), sing it using "tatata", hum it and whistle it (if he/she can whistle, of course).

#### 2.1.8 The end

At the end of the experiment, a word of gratitude for participating in the experiment is shown on the screen, a final message is played back and a possible response from the subject is recorded (could be used as an example of a non-query or noise recording). After that, the subject receives a small reward from one of the collaborators (a cinema ticket).

#### 2.1.9 Overview

In what follows an overview is given of the major steps in the experiment.

1. collecting info on the subject
2. collecting info on the subject's knowledge of the musical pieces
3. imitating known pieces without hearing them first (vocally, by whistling, or in any other possible way)
4. imitating pieces after hearing them in their entirety first (vocally or by whistling)
5. imitating a fixed fragment in 4 different ways (singing lyrics, singing "tatata", humming and whistling)

### 2.2 Input and output files

#### 2.2.1 Input files

The *configuration file* specifies the setup of the experiment. It is supplied as a parameter when running the experiment program and contains directory paths and 2 sets of entries for musical pieces. An example of a configuration file, using paths relative to the position where the executable is located can be found on the MAMI website (see below 'Access'). The example shows the configuration file used in the experiment described here, so all queries in our database are fragments that are related to these pieces.

Each entry for a piece contains the following information: a 3-digit ID of the piece (PID), names or words related to the composer, performer or another description, the title of the piece, the name of the sound file and the name of the text file containing the lyrics for the fragment (only used for Set5).

The *test sound files* for Set1 should be recordings of complete pieces and can be in any format supported by the used libraries. The names of the sound files are specified in the configuration file. We digitally recorded the pieces from the CD's in 44.1 kHz 16-bit stereo PCM WAV format and simply used the 3-digit piece ID's in the sound file names like this: PID.wav.

For the single fragment in Set5, a *lyrics file* is specified in the configuration file. This lyrics file contains the exact lines of text that are sung in the fragment.

#### 2.2.2 Output files

*Profile file*

This file is created for each subject and contains the following information: a unique ID for the subject, age and gender, the number of hours a week the subject actively listens to music, whether the subject plays an instrument or not (if so, the number of hours a week is added), the highest level of musical education (no musical education, music academy or music conservatory) and paths to the log file and profile file.

*Log file*

The log file keeps track of the course of an experiment for a specific subject (the subject ID is stored in the file name). Each log file consists of 4 parts, corresponding to the flow of

the experiment as described in the use case above. The following describes the output into the log file of each of these parts:

- Preparatory stage

  The log section for the preparatory stage of the experiment shows a division of pieces into 4 categories. Pieces were eventually classified as either "I known it and I can imitate it", "I don't know it", "(I think) I know it, but I don't remember how it sounds" or "I really do know it, but I can't imitate it".

- Experiment part 1

  The log section for experiment part 1 has multiple entries, one for each presented piece (specified by the piece ID). Each entry shows the file name(s) of the recorded vocal imitation(s) and is followed by an "other query" section. The latter can possibly contain any (or none) of the following in any order: a verbal query, a description of another query method, or the filename of an extra recording. There can be less than 10 log entries like this if the subject didn't specify enough pieces as "known".

- Experiment part 2

  Again, the log section has multiple entries, one for each presented piece (specified by the piece ID). Each entry shows whether the subject knew the piece after having listened to it, followed by an indication if this answer differs from the one in the preparatory stage. After that, the file name of the recorded imitation is shown, possibly followed by the file name of a second recording on the next line. There are always log entries like this for exactly 4 pieces.

- Experiment part 3

  The log section for this part shows an indication of whether the subject knew the fragment after having listened to it, followed by the number of times the subject listened to the fragment. Then, the file names of the recorded fragments are shown. Fragments are always recorded for singing with words, singing with syllables and humming. A fragment for whistling is only recorded if the subject indicated he/she could do that.

*Query sound files*

These are the files that are analyzed to investigate user preferences and that can be used for the evaluation of audio feature extraction algorithms. All generated query sound files are named consistently in a way that allows identifying the stage of the experiment where they were generated. There are sound files for the different query-by-voice trials in part 1 of the experiment (reproducing known pieces from long-term memory), the first recording and possibly one or two extra recordings (see above) and similarly for the 2nd part (producing queries after listening to the piece). In part 3, all subjects are asked to imitate a fragment heard using several methods: singing words, singing syllables, humming and whistling (not available if the subject indicated that she/he could not whistle it). This gives an extra 3 or 4 sound files per subject. A last sound file contains some spontaneous comments (if any) of the subject, recorded after the experiment has ended (mainly noise or laughter).

*Sound level check file*

This sound file is generated when a sound level check is performed (i.e. when running the program with the -L flag). It just contains the sound that was recorded the last time a sound level check was performed.

*Counter file*

This file always contains a single number representing the last used unique subject ID. Each time the program is run, this ID is incremented. The unique ID is used throughout the program for labeling the output files.

### 2.3 About the software itself

The application used for conducting this experiment was developed as a Win32 console application using MSVC++ 6.0 and was tested on Windows98SE and Windows2000 systems (we ran it on a Windows98SE machine for the experiment). Standard C++ was used as much as possible and the used libraries PortAudio (Bencina et al., 2001-) and libsndfile (de Castro Lopo, 1999-) are cross platform so it should be possible to build it on other platforms as well (possibly with minor modifications).

## 3 Database annotation

### 3.1 Annotators

Musicologists working at IPEM, Dept. of Musicology, Ghent University, annotated the MAMI query database. They are all experienced in computer assisted annotation using software tools such as Cool Edit, Pure Data and Matlab for assistance and verification.

### 3.2 Annotation strategy

The annotation strategy was focused on two objectives, one user-oriented, and one modeling-oriented. The user-oriented approach aimed at providing content about the spontaneous behavior of users taking part in the vocal query experiment. The model-oriented approach aimed at providing detailed descriptions of queries in order to build a referential framework for testing automatic transcription models.

User-oriented annotations were carried out for a set of 1148 queries taken from the user responses to the 30 songs used in the experiment. The user-oriented annotation focused on the analysis of long-term versus short-term memory effects, the different vocal methods used and the differences between subjects and their relation to musical education, age and gender.

The model-oriented annotations were carried out for 32 queries, excerpts of the popular songs “Blowin’ in the wind”, “Walk on the wild side”, “Sunday Bloody Sunday” and “My way”. These annotations provide detailed descriptions of low and mid level acoustical features, as described in the next section.

### 3.3 Annotated features

#### *3.3.1 User-oriented annotation*

The user-oriented annotation was performed on the following features: timing, query method, performance style, target similarity, and syllabic structure. In the field of timing, the total length of the recording, the start, end and length of the actual query were collected. Vocal queries may contain a mix of different query methods such as humming, singing syllables, singing lyrics and whistling. In addition to those methods percussion (e.g. tapping along with the drum) and comments (spoken comments made by the subjects) are found. Studying these six methods in more detail required segmentation into homogeneous parts. Segmentation in temporal units according to the methods used resulted in a set of 2114 segments.

Segment annotations at the temporal level indicate the starting and ending time as well as the total duration of the segment. The number of segments in a query (according to the used methods) is counted. Furthermore, distinction is made between three different performance styles accentuating on melodic, rhythmic or intermediate interpretations. A performance style is considered to be melodic when a clear succession of different pitches, or melodic intervals is observed. A segment is annotated as rhythmic when no clear pitch intervals are noticed (as in a spoken text or a percussive sequence). An intermediate category is used to classify segments where a sense of pitch is present, but without a clear melody (e.g. using a reciting tone). Then, to each of the segments, a relative similarity rating is given on a six-point scale, ranging from not recognizable (0) to sounding similar (5) to the target song (presented textually in part 1 and aurally in part 2). The estimation of the degree of similarity between query segment and target is focused on melodic and rhythmic properties. Aspects of timbre or use of lyrics are neglected. This estimation, obviously, is subjective and therefore the similarity measures are only used to compare large sets of data. That is to compare the efficiency of the different methods, the performance of different groups of users and the effects of differences in memory recall.

As 766 segments out of 2114 have a syllabic content, a major part of the annotation work is related to the analysis of syllabic queries. Syllables are considered as non-semantic vocal events, containing a vowel, which is preceded and/or followed by a consonant or a complex of consonants. Syllables are analyzed according to their structural components: the onset (initial consonant or complex of consonants), the nucleus (vowel) and the coda (final consonant). The 766 syllabic segments in the database contain a total number of 14748 syllabic units.

#### *3.3.2 Model-oriented annotation*

The model-oriented annotation is based on a set of homogeneous query segments containing singing syllables or whistling, and a set of heterogeneous queries containing a mixture of methods. The features investigated are event, onset, frequency, pitch stability, method and sung words or syllables.

At the local temporal level, events are determined. An event or object is characterized by its duration. It starts at the moment in time defined by the beginning of an onset and ends at the moment in time where the onset of the next event or non-event begins. From a conceptual point of view, an onset is considered a moment in time defined by the beginning of an event. This definition accounts for successive events pertaining to monophonic files. The point of onset is based on both auditory (listening with headphones) and visual (looking at the waveform) perception. Moreover, a sureness quotation is given (0, 1) which distinguishes between clear onsets (1) and less pronounced onsets (0). An onset is considered less pronounced when successive notes with the same or different pitch are performed smoothly with no explicit separation (e.g. legato performance).

For pitch annotation, frequency was assigned in Hertz with a resolution of one Hertz. Taking into account that the query files in the data set are real audio and not MIDI encoded a resolution greater than a semitone makes sense. It facilitates adaptation to users who sing too high or too low with frequency deviations that do not coincide with a semitone.

### 3.4 Method of model-oriented annotation

For model-oriented annotation the PRAAT program for speech analysis (Boersma & Weenink, 1996) is used. Labeling and segmentation of the sound files is stored in a TextGrid object that is separated from the sound. Boundaries are marked by the places in time where an event, a non-event or a pause for breath begins. The TextGrid object is written into a formatted ASCII-text file. Time points are labeled on multiple tiers that are time synchronized for different annotation levels as shown in figure 1.

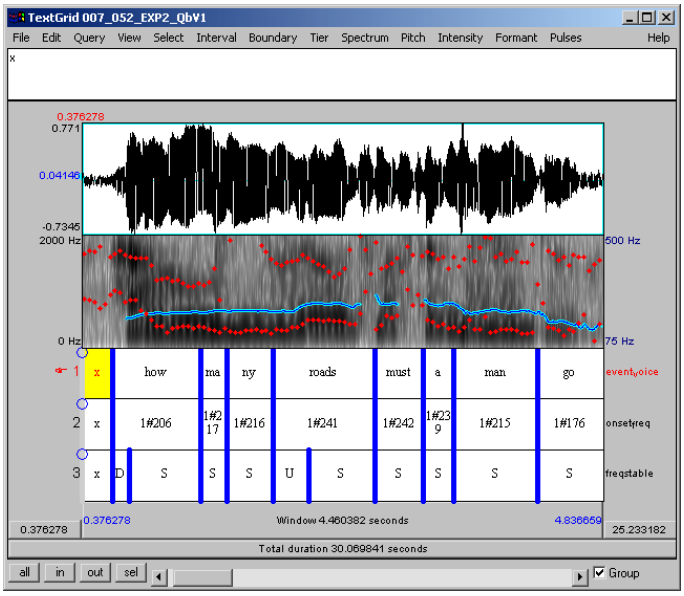


Figure 1: Vocal query annotation using PRAAT. The tiers from top to bottom show (1) the sound waveform, (2) the pitch contour of a sound as a function of time, (3) used lyrics or query method, (4) sureness quotation and frequency and (5) pitch stability.

Tier objects representing non-events such as breathing or a clapping door are labeled with an 'x'. The query method is labeled with h (humming), p (percussion), w (whistling) or c (comment). For sung queries the lyrics or nonsense syllables have been annotated. Frequency estimation was done aurally, directly comparing the sung tones with pure tones generated by a frequency generator implemented in Pure Data (Puckette, no date). Then a final check is done, reproducing the annotated frequencies simultaneously with the original sounds, using a Matlab script. To each event, an extra segmentation descriptor has been added defining pitch dynamics in terms of stability. This feature refers to the steady part of a note where all the harmonics become stable and clearly marked in the spectrum. Stability annotation is conforming to four semantic labeled categories: up, down, stable and fluctuating. A pitch is considered stable when the distance between the lowest and highest frequency within an event is equal to or smaller than 5 Hz. As sung melodies, especially when produced by untrained voices, move within a limited frequency range, this threshold rarely exceeds a quartertone.

The model-oriented database was used as a reference framework for testing pitch to MIDI models. Results of this investigation are discussed by Clarisse et al. (2002).

## 4 Overview of the user-oriented annotation

Statistical analysis of the user-oriented annotation provided insight in the structure of a query-by-voice search. Some basic characteristics of vocal queries and several user categories could be distinguished (Lesaffre, Moelants & Leman, in press; Lesaffre et al., in press).

### 4.1 Timing

Analysis of the timing characteristics of the queries shows a mean query length of around 14 seconds while the actual query starts 634 ms (average) after the start of the recording. But the between-subjects variance is considerable.

### 4.2 Query methods

Six query methods are distinguished, most common are singing lyrics and singing syllables. Whistling is the third most popular method, while actual humming, percussion and comments occupy only a small share of the whole. In 40% of the queries a mixture of at least two methods is used. Also the use of certain methods is shown to be user dependent. Five user categories have been distinguished that (1) show preference for singing text, (2) for singing syllables, (3) alternate between two methods, (4) mix several methods, and (5) show preference for whistling. Around 75% of the queries are performed in a melodic way.

### 4.3 Syllable structure

In the detailed analysis of the syllabic queries a total of 23 different onsets and 37 rhymes is found, but 99.3% is organized in 11 onsets and 19 rhymes. The ten most common syllables are (in order of decreasing importance) [na], [n@], [la], [t@], [da], [di], [d@], [ta], [tu] and [ti], of which [t@], [na] and [d@] belong to the syllable repertoire of the largest number of subjects. Syllables are transcribed using SAMPA (http://www.phon.ucl.ac.uk/home/sampa/home.htm) that is a machine-readable variant on the International Phonetic Alphabet. Analysis of the type, spread and clustering of the syllables also distinguished between different user categories. For onset as well as for nucleus different user groups were found. Users may use specific onsets ([n] and [l]) or nuclei ([a] and [@]). Some users clearly prefer the onset cluster [d-t-r] while others go for the alternation of [a] and [@] nuclei.

### 4.4 Effects of age, gender, musical experience

Significant effects of age, gender and musical backgrounds were found. It is shown that younger people tend to start their queries sooner and have a better similarity score. Men tend to start their queries later than women do and use a larger variety of syllables with a larger share of onset [t] and a smaller amount of [a] nuclei. Musicians make longer queries than non-musicians and use less text (in favor of syllables and whistling) than non-musicians. The use of [a] nuclei and [l] onsets and of comment as a query method increases with age.

### 4.5 Memory

Differences were also found between the reproduction of unfamiliar melodies from short-term memory and the recall of known melodies from long-term memory. The timing of the queries as well as their similarity with the target is dependent on the type of memory used by the subjects. When known songs are 'refreshed' by presenting them aurally, the queries start sooner and last longer. The reproduction of unfamiliar melodies from short-term memory has less quality than recall of known melodies even if the query only relies on long-term memory. The lesser degree of acquaintance is also reflected in the larger share of syllabic segments and the increased importance of rhythmic and intermediate performances.

## 5 Access

The database can be accessed from the public section of the MAMI project website: http://www.ipem.ugent.be/MAMI. The website also provides more detailed descriptions of the setup, the input and output files, the queries and annotations, its availability and links to electronic versions of related papers.

## 6 Conclusion

Annotation of music collections is often seen as a necessary step for the development of MIR systems. Byrd (2003) has for example described candidate music IR test collections, their characteristics and availability. However none of these collections contain human produced queries stored as audio. Researchers tend to work with databases containing small sets of queries with tunes obtained from small-scale experimental studies. Query sets with sung tunes, are far from elaborate and there is a need for thorough studies related to the user's singing preferences and habits (Downie, 2003). This paper presented an elaborate experiment that generated a query database with melodies that represent all possible vocal query methods. Investigations concerning user behavior and system

development show that the use of manually annotated files yields a better conceptual understanding of some of the recurring issues encountered in an undertaking of musical content retrieval. The availability of the MAMI vocal query database and its annotations provides basic material for MIR research.

**Acknowledgements**

The authors wish to thank Jelle Dierickx and Liesbeth De Voogdt for their assistance in annotating the query files. We thank Johannes Taelman for his assistance in working with Pure Data. This research has been conducted in the framework of the MAMI project for audio recognition at IPEM, Department of Musicology at Ghent University. The Flemish Institute for the Promotion of Scientific and Technical Research in Industry gives financial support for this project.

**References**

Bencina, R., Burk, P., Dannenberg, R., McNab, D., Eldridge, B., et al. (2001-). *PortAudio, a free, cross platform, open-source, audio I/O library*. Retrieved July 31, 2003, from http://www.portaudio.com.

Boersma, P., & Weenink, D. (1996). *Praat. A system for doing phonetics by computer*. Amsterdam: Institute of Phonetic Sciences of the University of Amsterdam. Retrieved July 31, 2003, from http://www.praat.org.

Byrd, D. (2003). Candidate Music IR Test Collections Retrieved July 31, 2003, from http://php.indiana.edu/~donbyrd/MusicTestCollections.HTML

Clarisse, L. P., Martens, J.-P., Lesaffre, M., De Baets, B., De Meyer, H., & Leman, M. (2002). An Auditory Model Based Transcriber of singing sequences. In M. Fingerhut (Ed.), *Proceedings of the Third International Conference on Music Information Retrieval* (pp. 116-123). Paris: IRCAM – Centre Pompidou.

de Castro Lopo, E. (1999-). *libsndfile, a C library for reading and writing files containing sampled sound through one standard library*. Retrieved July 31, 2003, from http://www.zip.com.au/~erikd/libsndfile.

Downie, S. J. (2003). Music information retrieval. *Annual Review of Information Science and Technology*, *37*, 295-340. Retrieved July 31, 2003, from http://music-ir.org/downie_mir_arist37.pdf.

Leman, M., Clarisse, L. P., De Baets, B., De Meyer, H., Lesaffre, M., Martens, G., Martens, J.-P., & Van Steelant, D. (2002). Tendencies, Perspectives, and Opportunities of Musical Audio-Mining. *Journal Revista de Acústica*, *XXXIII*(3-4), abstract p. 79, full text: CD-ROM.

Lesaffre, M., Moelants, D., & Leman, M. (in press). Spontaneous user behavior in "vocal" queries for music-information retrieval. *Computing in musicology*.

Lesaffre, M., Moelants, D., Leman, M., De Baets, B., De Meyer, H., Martens, G., & Martens, J.-P. (in press). User behavior in the spontaneous reproduction of musical pieces by vocal query. In: *Proceedings of ESCOM5*. Hannover.

Puckette, M. S. (no date). Pure Data. Retrieved July 31, 2003, from http://www.pure-data.org/.