MTG-DB: A Repository for Music Audio Processing

Pedro Cano, Markus Koppenberger, Sira Ferradans, Alvaro Martinez,
Fabien Gouyon, Vegard Sandvold, Vadim Tarasov, Nicolas Wack
Music Technology Group - Institut de l'Audiovisual
Universitat Pompeu Fabra
Ocata 3, 08003 Barcelona, Spain
pcano@iua.upf.es

June 8, 2004

Abstract

Content-based audio processing researchers need audio and its related metadata to develop and test algorithms. We present a common repository of audio, metadata, ontologies and algorithms. We detail the hardware implementation, in the form of massive storage and computation cluster, the software and databases design and the ontology management of the current system. The repository, as far as copyright licenses allow, is open to researchers outside the Music Technology Group to test and evaluate their algorithms.

1 Introduction

Developing technologies related to musical audio signal processing requires data. For instance, implementing algorithms for automatic instrument classification requires annotated samples of different instruments. Implementing a voice synthesis and transformation software calls for repositories of voice excerpts sung by professional singers. Testing a robust beat-tracking algorithm requires songs of different styles, instrumentation and tempi. Building models of musical content with a machine learning rationale calls for large amounts of data. Besides, running an algorithm on big amounts of diverse data is a requirement to ensure its quality and reliability. The main purpose of this paper is to describe the research infrastructure in place now in the Music Technology Group of the Universitat Pompeu Fabra:[1] the MTG-Database (MTG-DB). The MTG-DB is a common database of audio material that offers functionalities for adding audio content, browsing the database, adding metadata and dealing with taxonomies and algorithms.

The MTG-DB is not a system for information retrieval evaluation. Its goal is not to stand as a benchmark framework like the one pursued in the MIR community [9].

[1] `http://www.iua.upf.es/mtg`

The MTG-DB is rather concerned with providing a common storage for audio material and associated metadata (see Section 2) as well as tools for research (see Section 3). Section 4 describes the system architecture. Section 5 details the actual audio content and comments on copyright issues.

2 A common repository

Too often researchers build their audio collections from scratch and annotate corresponding metadata to develop and evaluate algorithms. In order to facilitate reuse of this material, the MTG-DB centralizes in a common repository audio content, corresponding metadata, taxonomies and algorithms.

2.1 Audio

A common repository of audio allows to confront algorithms with big repositories of different nature. As outlined in the introduction, machine learning approaches to music audio processing require large amounts of data. Livshin *et al.* point out in [11] that evaluation of sound classifiers on a single database is not necessarily an indicator of the generality of the classifier nor its suitability for practical applications. Besides the application to the development of data-driven algorithm, large collections of audio have more benefits. The simple act of running an algorithm against a database much bigger than the one used for development can reveal a lot of information about the algorithm. The algorithm is tested in robustness, both on the accuracy with very distinct data and on the soundness of the actual implementation, e.g: algorithms designers may have not contemplated a certain situation that makes a program crash.

2.2 Metadata

There are many advantages in centralizing metadata together with the audio content. The metadata and possible descriptions linked to the audio content allow users to compare their algorithm with others'. Different researchers working on the same problem can increase the benchmark database size by adding up more ground truth data.

The common repository can help discover synergies between automatically extracted descriptors. For instance, someone studying chord segmentation will benefit from the results of a beat tracker. Somebody studying the structure of music is likely to benefit from results from tempo evolution, chord progression and singing voice detection.

A centralized and correctly labeled repository can stimulate corpus-oriented research and the use of statistical methods and learning techniques. For instance, finding correlations between certain low-level descriptors and genres, or studying the performance of a key extractor in data of different classical periods, from baroque to romanticism, is made much easier.

2.3 Taxonomies

Along with metadata, it is important to store taxonomies or ontologies. The development of a percussive instrument classifier requires the definition of a taxonomy or classification scheme of the categories of percussive instruments. Same happens in many other cases: taxonomies of musical genres, voice types, singing styles, playing modes and many more. Someone spending a Ph.D. thesis time researching on a specific topic is likely to have organized a taxonomy or an ontology that describes the concepts of that specific topic. It is imperative to preserve that knowledge. Much of the legacy metadata may refer to terms of such ontologies. If we want to reuse the metadata, we need to know exactly which restricted vocabulary was used and what was the meaning of each term. Metadata has to be well specified so that it is comprehensible both by men and computers alike. The following is an example of possibly cryptic textual metadata that describes an acoustic guitar chord: "AcousticGuitar:BbMin:Hi:Down". In order to describe this audio sample, besides specifying the instrument that produce it—an acoustic guitar—it is interesting to encode somehow that acoustic guitar is a type of string instrument. It is important to specify that the sound sample is a chord and whether it was created strumming up or down, type of strum, and so on. Different ontologies are managed with an extended version of WordNet.[2] See Section 3.2 and [7] for details.

2.4 Algorithms

Different algorithms are stored with associated documentation describing its use, under which operating systems they have been tested, who is maintaining them, relevant publications and assumptions on the operating conditions. It is possible to upload and register algorithms developed in any of the common programming languages; yet, to take full advantage of the framework, like using the high performance cluster environment (see Section 4.2), they have to run on GNU/Linux. Many algorithms use the CLAM[3] framework, a C++ Library for Audio and Music, developed at the MTG and distributed under the GNU General Public License.[4] It is relatively easy to build a complex algorithm using a number of previously registered algorithms.

3 Research Tools

A big and centralized repository of audio and metadata improves the robustness and accuracy of the audio technologies. However it takes some effort for the researchers to put the metadata and the algorithms in a compliant format that can be easily registered in the system. In order to attract researchers into populating and using the repository, the system offers functionalities such as tools for metadata annotation, ontology management, an information retrieval framework and an experimental framework.

[2] `http://www.cogsci.princeton.edu/~wn`
[3] `http://www.iua.upf.es/mtg/clam`
[4] `http://www.gnu.org/copyleft/gpl.html`

3.1 Annotation tools

Textual annotation for describing audio content — such as artist, genre or mood of a music piece — as well as ontology management is provided via a web interface. Some descriptions however refer to segments of audio, for example the structure of a song: Intro, Bridge, Chorus, the chord progression, the instruments played at certain parts or solo parts. In these cases, the audio segments are annotated using Kåre Sjölander and Jonas Beskow's WaveSurfer.[5] WaveSurfer is an open source tool for sound visualization, annotation and transcription. We have extended the functionality of this software in several aspects. It is possible to launch the application from within the web interface. The audio is loaded automatically along with segmentations already available on the repository. A connection to the MTG-DB server via web services allows to download, edit or upload new segmentations. The web service validates the labels used for tagging the audio against a restricted vocabulary. Assuring annotation consistency is important, especially if several people are contributing to the repository to ensure that, for example, "Intro" when describing parts of a song is always labeled as "Intro" and not "intr", "introduction". View Figure 1 for a screenshot of WaveSurfer.

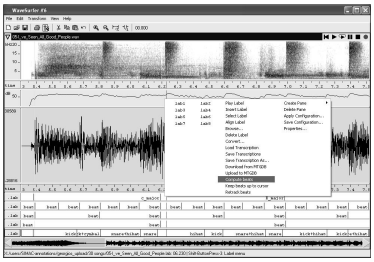

Figure 1: WaveSurfer annotation tools screenshot

3.2 Ontology management tools

According to Wikipedia, taxonomy may refer to either a hierarchical classification of things, or the principles underlying the classification.[6] In computer science, an ontology is the attempt to formulate an exhaustive and rigorous conceptual schema within a given domain, a typically hierarchical data structure containing all the relevant entities and their relationships and rules (theorems, regulations) within that domain.[7] The use of taxonomies or classification schemes alleviates some of the ambiguity problems inherent to natural languages. The MPEG7 standard provides description mechanisms and ontology management tools for multimedia documents[13]. We have decided to extend existing semantic network, WordNet, rather than starting from scratch. WordNet [15] is a lexical network designed following psycholinguistic theories of human

[5] `http://www.speech.kth.se/wavesurfer`
[6] `http://en.wikipedia.org/wiki/Taxonomy`
[7] `http://en.wikipedia.org/wiki/Ontology`

lexical memory. Standard dictionaries organize words alphabetically. WordNet organizes concepts in synonym sets, *synsets*, with links between the concepts like: broad sense, narrow sense, part of, made of and so on. It knows for instance that the word "piano" can refer to the musical attribute that refers to "low loudness" and the musical instrument. It also encodes the information that a "grand piano" is a type of piano, and that it has parts such as a keyboard, a loud pedal and so on. For details on how the framework deals with the ontology management, we refer to [7].

3.3 Information retrieval tools

A great deal of music audio processing research is currently devoted to music information retrieval. Some algorithms extract sonological and musical descriptions from audio that can be used for providing new ways of interacting with large collections of music. The MTG-DB provides a framework for searching, browsing and displaying results. In this framework, a user registers a compliant description extractor, the extractor is run on a selection of audio files, the extracted descriptors are stored in the repository and, automatically, tools are available for:

1. Query by description: It applies when the metadata is in textual form, which includes labels such as mood, genre, duple/triple meter or key.

2. Query by numerical values: There is a slider interface for numerical descriptors such as beats per minute, loudness, percussiveness (see Figure 2).

3. Query by example: The user uploads an audio file and asks for similar audio items.[8]

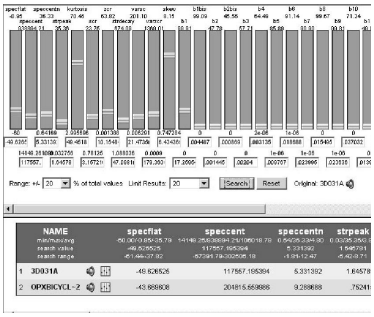

Figure 2: Information retrieval slider interface

There are also tools for visualization and playback in different modes.

[8]Certain descriptors require to register in the system the similarity measure that reads the descriptors and is capable of comparing and ranking them. For example, an extractor can output a representation of a song in the form of a Gaussian Mixture Model of Mel Frequency Cepstrum coefficients [2]. In order to rank songs given such a descriptor, a helper application has to be registered in the system.

1. Flat list: Flat list of results ranked by relevance, as in Google[9] or Altavista.[10]

2. Category tree: Browsing through categories trees is available in two ways: First using the registered ontologies—musical genres, musical instruments, voice types—and, second, via hierarchical clustering over the extracted descriptors [6].

3. 2D Representations Maps: Multidimensional scaling-like visualization that projects (dis)similarity distance matrices onto an 2D Euclidean representation [5].

This set of information retrieval tools alleviates the workload of the researchers. Instead of having to design prototypes for demonstrations, they can concentrate on their algorithms.

3.4 Connection with external software

Web-based technologies, including XML and Web services, allow the seamless interoperability with external applications (see Section 4). Specifically there is a connection with Matlab.[11] Web services provide as well an interface to the annotator (see Subsection 3.1) and to audio players: Winamp[12] and XMMS.[13]

The XML interface together with XSLT is used both for the web interface (see Figure 3) and also to create data compliant with machine learning frameworks, in particular Weka.[14]

3.5 Data-mining tools

Beside audio, metadata, ontologies and algorithms, it is possible to store a history of classification experiments. In this setup, each classification experiment consists of a subset of audio data, several algorithms with accompanying configuration files and algorithm outputs. Experiments are stored with a descriptive name and date for later retrieval.

By a simple one-step process any experiment can be re-run, with possible changes to data or configuration. The new results are then stored separately.

By keeping track of all experiment results, the evolution in performance can be monitored. By looking for maxima or minima on graphs, successful experiment configurations and/or trends are readily identified. Manual and/or subjective evaluations can also be done, since all results are available for download. This is an important issue, since it is not feasible to implement a system that will satisfy everyone's needs in particular. To avoid limiting the work process of the individual, all parts of the experiment procedure must be able to perform offline.

[9] `http://www.google.com`
[10] `http://www.altavista.com`
[11] `http://www.mathworks.com`
[12] `http://www.winamp.com`
[13] `http://www.xmms.org`
[14] `http://www.cs.waikato.ac.nz/ml/weka`

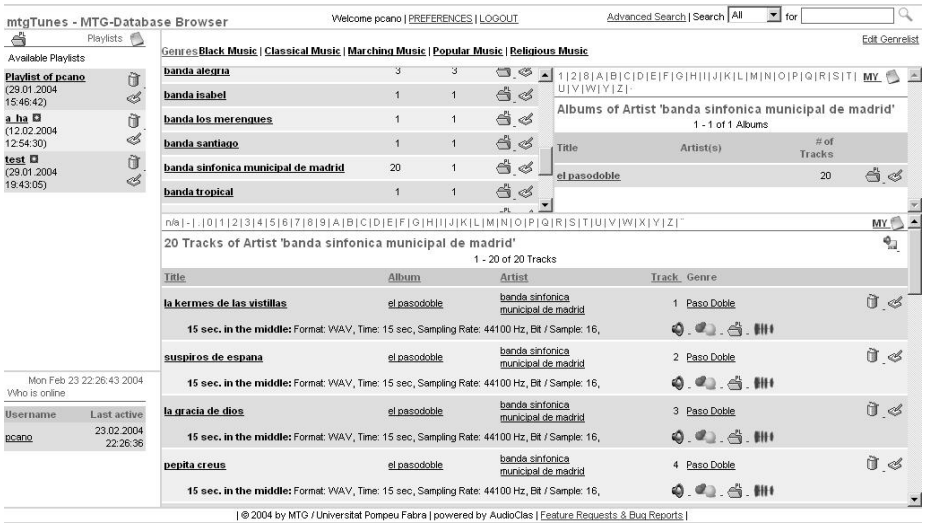

Figure 3: Web interface

3.6 Robustness test

Algorithms should ideally be robust to different quality versions of the same content. The consistency of an algorithm against distortions such as low-pass filtering, re-sampling, background noise, mp3 or GSM transcoding is an indicator of the quality and generality of the approach. This testing and some other functionalities are implemented in the MTG-DB framework so as to avoid duplicating work and promote a faster and higher quality research.

4 System Description

All the tools and functionalities are independent of O.S. and database architecture. There is a XML Interface that allows reusing the framework within a stand-alone PC application or web interface with different skins (see Figure 4).

4.1 Block Diagram

DATA STORAGE: The audio data is stored in a normal file system. Information about the files and metadata are saved in a relational database. Some metadata, such as MIDI or low-level descriptors, may also be stored in the file system. These files are accessible via a link in the database.

APPLICATION SERVER: The main core functionalities reside on the application server. It allows to access all the data—audio, metadata, algorithms and taxonomies—stored in the repository. The file information and metadata are represented as objects and access to the audio files is available in different ways: http-download, direct access

via NFS or Samba in the intranet. Registered users can add new data and modify existing data, provided they have sufficient access rights. Standard search tools are provided directly by the application server. This includes full-text and specific text searches— e.g. search only in title-field of tracks—and, more generally, the types of searches described in Subsection 3.3.

Users can define virtual collections or play lists in two different ways:

1. Manual Playlists: Users can create playlists and add objects—files, albums, artists—to them. These lists will be stored in the system for later retrieval, e.g: to re-run an experiment with the same files.

2. Automatic Playlists: The results of a search can become a playlist, e.g: "Find all reggae pieces with more than 90 bpm". Search parameters can be saved for later re-use. There is also the possibility to manually modify such a playlist.

Built-in conversion of audio data from and to the common formats (wav, mp3, aiff) with optional parameters such as re-sampling is available. Conversion can be done on single files, collections or the whole repository. Besides downloading whole audio files, segments can be created on the fly and downloaded as different files, e.g: "Give me all sax solos".

CLIENT ACCESS: There are two possible ways of client access:

1. A web interface, which is accessible by common browsers, allows to use most of the functionality (e.g: browsing, searching, editing editorial meta data, ...)

2. All functionality can be exported via a SOAP Interface. The SOAP Interface provides some exclusive functionality such as interaction with specific applications, e.g: WaveSurfer, which are not be available via the web interface. SOAP[15] is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. The service interface is described in a WSDL[16] document indicating methods calls, objects, and exceptions that will be sent across the net. As all the exchange of information is made with XML (both data and control messages), SOAP makes the interaction between different platforms and programs running anywhere on the Internet possible.

IMPORT/EXPORT: Different tools to add new collections (e.g: CD's) are provided. In the MTG-intranet audio-data can be uploaded via NFS/Samba into a writable import directory and registered to the repository using the web interface. For larger collections several command-line scripts can be used by the administrators of the system to do a semi-automatic import. The import scripts perform several name entity analysis so as to ensure avoid duplicates in the editorial metadata[14, 1]. The scripts use soundex and string matching algorithms [14], as well as some heuristics and cross-checking of information with FreeDB[17] and MusicBrainz.[18]

Every list of files / tracks — that also includes search results or playlists — can be exported, that is, made available for download. Different forms of export are available:

[15] `http://www.w3.org/TR/soap`
[16] `http://www.w3.org/TR/wsdl`
[17] `http://www.freedb.org`
[18] `http://www.musicbrainz.org`

1. The user can download a list of files as plain text (or m3u files) to work on these files on the mounted repository (intranet-only).

2. The audio-data and specified metadata is collected in a temporary directory, packed and compressed and — using the web-interface — the user is provided with a link to download the package.

4.2 Hardware setup

The MTG-DB requires a great deal of storage and high performance computing capabilities. Shortly the infrastructure consists of:

1. Massive and High-Availability Storage: The common repository requires a big amount of hard disk space. At the moment of writing the audio is stored in a 4 Terabytes RAID 5.

2. Distributed computation cluster: Besides the possibility of selecting a list of audio items and downloading them in a researcher's PC for processing, the use of the cluster is considered. In such a setup, the researcher uploads the executable that processes the batch of files and the computation is done in the distributed environment. The computational cluster consists of 16 nodes (AMD Dual 2Ghz) and a master (AMD Dual 2Ghz).

3. Servers: There are two servers connected to the RAID through a dumb file-server. On one of the servers resides the application server. The second one controls user access rights and exports the audio collection via NFS and SAMBA.

4.3 Content-based Metadata Specification

Agreeing on which metadata should be included in the database scheme is a daunting task. Support is given to three types of metadata: textual, numeric and everything else.

1. Textual descriptors or labels.

2. Numeric descriptors: scalar or vectors, such as Loudness, BPM, speech/music factor.

3. Arbitrary representations: musical score, lyrics, hidden markov models. The similarity measure that can deal with arbitrary representations—for a query by example type of approach, a clustering algorithm or a visualization—is provided as long as the tools to compare—or to search within—arbitrary representations are provided as well. This design is meant to avoid the incredible work load that could derive from trying to find a comprehensive description framework for audio data—both for the definition of such a framework and the implementation.

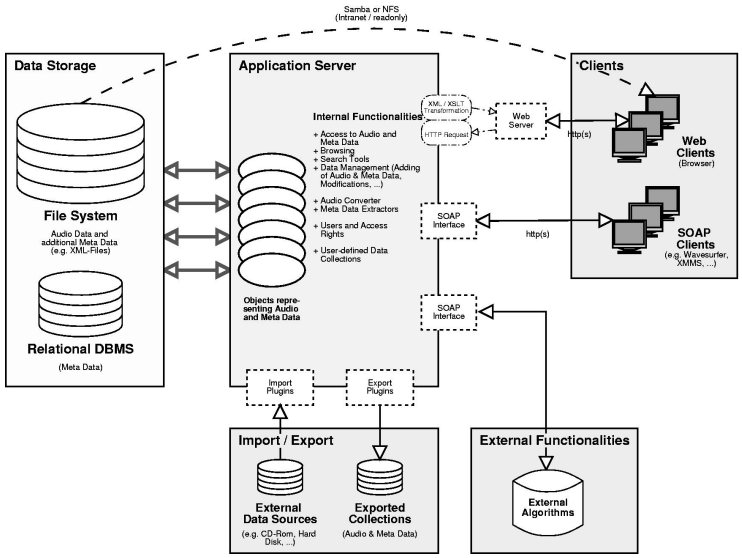

Figure 4: Block Diagram

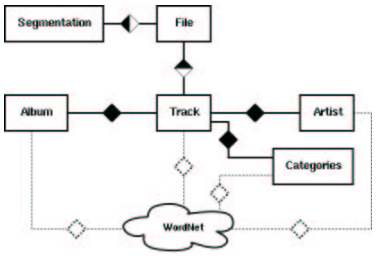

Figure 5: Simplified DB Schema

5 Miscellaneous

5.1 Audio data

Over the years and as a result of different projects we have gathered a substantial amount of heterogeneous audio content. As of February 2004, there were:

1. Musical instrument samples and sound effects.
2. Monophonic phrases with different levels of expressiveness.
3. Drum loops
4. Singing voices: Different voice qualities and expression.

5. Music: over 80.000 titles covering a wide variety of genres and over 9000 different artists.

Some parts of this collection are copyright protected and therefore cannot be accessed from outside. Some other collections have a different licensing terms (see Subsection 5.2) that allow distribution, such as the University of IOWA Music Instrument Sample Database.[19] Another important collection is the RWC. The RWC (Real World Computing) Music Database is a copyright-cleared music database (DB) that is available to researchers as a common foundation for research [12].

With respect to the quality of the audio, the premises are clear: obviously the higher the quality of the audio the better (at least stereo 44100 Hz). Some of the audio content is not of high quality yet they have good descriptions. In these cases, the audio is stored in the system along with a tag that indicates that is low-quality audio. The search engine allows to filter out the results according to quality.

5.2 Copyright Issues

Researchers from outside our lab can access the repository. Some audio content, however, is copyrighted. In this case, like it has been proposed by [2], users will only have access to metadata. Together with the audio editorial description, the users can download a set of precalculated descriptors of common use—some included in the MPEG7 standard [13]—, such as Mel-Frequency Cepstrum, spectral flatness, BPM, prominent-pitch.

Nevertheless, there is still full access to not "all-rights-reserved" type of copyright. The common creatives license[20] defines alternative licensing schemes. The MTG-DB currently provides metadata and pointers to the content in the following sites: www.magnatune.org, www.opsound.org.

5.3 Disclaimer

The goal of the system is not to offer an information retrieval benchmark framework like TREC or OpenVideo, like it is being pursued in the Music Information Retrieval community [9, 16] for two reasons:

1. We do not provide evaluation retrieval tools.

2. We are interested in other aspects besides information retrieval, such as content-based music processing. In that area, there are many subtasks that can be evaluated: the performance of beat tracking, prominent pitch, structure detection, spectral analysis-resynthesis quality [10]. It would be out of our capabilities to design and implement all possibilities. That is why the metadata is just linked to audio content and to the algorithm relevant to it. When a given researcher wants to compare or improve an algorithm, they can get all information easily and duplicate whatever experiment to compare results.

[19] `http://theremin.music.uiowa.edu/MIS.html`
[20] `http://www.commoncreative.org`

It is more similar in a way to the UCI Repository of machine learning databases [3]. The purpose is not to end-up like an editor company with plenty of information like who has got the copyright or where was it recorded but the descriptors that are relevant for content-based processing research.

6 Summary

We have introduced a large-scale repository for music audio content based processing that centralizes audio content, metadata, ontologies and algorithms. It aims at stimulating higher quality research, by evaluating algorithms on a big and diverse corpus. The common repository allows duplicating experiments as well as sharing both experimental results and procedures. The MTG-DB repository is accessible for researchers outside the MTG Lab as far as copyright licenses allow. The repository is accessible at www.mtgdb.org.

7 Acknowledgments

Part of this work has been funded under the AUDIOCLAS EUREKA PROJECT E!2668. We would like to thank Eloi Batlle, Oscar Celma, Maarten de Boer, Emilia Gómez, Enric Guaus, Perfecto Herrera, Jaume Masip and Miguel Ramirez.

References

[1] S. Baumann, A. Klüter, and M. Norlien. Using natural language input and audio analysis for a human-oriented MIR system. In *Proceedings of the WEDELMUSIC*, Darmstadt, Germany, 2002.

[2] A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman. A large-scale evalutation of acoustic and subjective music similarity measures. In *Proceedings of the International Symposium on Music Information Retrieval*, Baltimore, Maryland, 2003.

[3] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[4] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems, in press*.

[5] P. Cano, M. Kaltenbrunner, F. Gouyon, and E. Batlle. On the use of FastMap for audio information retrieval. In *Proceedings of the International Symposium on Music Information Retrieval*, Paris, France, 2002.

[6] P. Cano, M. Koppenberger, S. L. Groux, P. Herrera, and N. Wack. Perceptual and semantic management of sound effects with a WordNet-based taxonomy. In *Proc. of the ICETE*, Setúbal, Portugal, 2004.

[7] P. Cano, M. Koppenberger, P. Herrera, and O. Celma. Sound effects taxonomy management in production environments. In *Proc. AES 25th Int. Conf.*, London, UK, 2004.

[8] L. de C. T. Gomes, P. Cano, E. Gómez, M. Bonnet, and E. Batlle. Audio watermarking and fingerprinting: For which applications? *Journal of New Music Research*, 32(1), 2003.

[9] J. S. Downie. Toward the scientific evaluation of music information retrieval. In *Proceedings of the International Symposium on Music Information Retrieval*, Baltimore, Maryland, 2003.

[10] P. Herrera. Setting up an audio database for music information retrieval benchmarking. In *Proceedings of ISMIR 2002 - 3rd International Conference on Music Information Retrieval*, 2002.

[11] A. Livshin and X. Rodet. The importance of cross database evaluation in sound classification. In *Proceedings of the International Symposium on Music Information Retrieval*, Baltimore, Maryland, 2003.

[12] T. N. M. Goto, H. Hashiguchi and T. Oka. RWC music database: Music genre database and musical instrument sound databases. In *Proceedings of the International Symposium on Music Information Retrieval*, Baltimore, Maryland, 2003.

[13] B. S. Manjunath, P. Salembier, and T. . Sikora. *Introduction to MPEG-7. Multimedia Content Description Interface*. John Wiley & Sons, LTD, 2002.

[14] A. Martinez, S. Ferradans, M. Koppenberger, and P. Cano. Natural language tools applied to music information processing. *MTG2004-2 Internal Report*, 2004.

[15] G. A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, pages 39–45, November 1995.

[16] J. Reiss and M. Sandler. MIR benchmarking: Lessons learned from the multimedia community. In *Proceedings of ISMIR 2002 - 3rd International Conference on Music Information Retrieval*, 2002.