Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE4.2.1
Content Indexing, Monitoring and Querying

Version: 1.7
Date: 12/10/2005
Responsible: FHGIGD (revised and approved by DSI)

Project Number: IST-2-511299
Project Title: AXMEDIS
Deliverable Type: Report and Prototype
Visible to User Groups: Yes
Visible to Affiliated: Yes
Visible to Public: Yes

Deliverable Number: DE4.2.1
Contractual Date of Delivery: M13 (end of September 2005)
Actual Date of Delivery: 10/Nov/2005
Work-Package contributing to the Deliverable: WP4.2
Task contributing to the Deliverable: WP4.2
Nature of the Deliverable: Report and Prototype
Author(s): FHGIGD, DIPITA, DSI, EPFL, EXITECH, UNIVLEEDS

Abstract:
This deliverable aim is to show the research work done in the area of content indexing, monitoring and querying. It also reports the prototypes implemented to demonstrate the research work done in the different areas studied. The prototypes presented will be used inside the WP5, AXMEDIS Framework, WP9, Demonstrators and, obviously, in the continuation of WP4.

Keyword List:
Content Indexing, Crawler, Database, Querying, Content Descriptor, Fingerprinting

Table of Contents

AXMEDIS Copyright Notice

The following terms (including future possible amendments) set out the rights and obligations licensee will be requested to accept on entering into possession of any official AXMEDIS document either by downloading it from the web site or by any other means.

Any relevant AXMEDIS document includes this license. PLEASE READ THE FOLLOWING TERMS CAREFULLY AS THEY HAVE TO BE ACCEPTED PRIOR TO READING/USE OF THE DOCUMENT.

1. **DEFINITIONS**
   - i. "**Acceptance Date**" is the date on which these terms and conditions for entering into possession of the document have been accepted.
   - ii. "**Copyright**" stands for any content, document or portion of it that is covered by the copyright disclaimer in a Document.
   - iii. "**Licensor**" is AXMEDIS Consortium as a de-facto consortium of the EC project and any of its derivations in terms of companies and/or associations, see www.AXMEDIS.org
   - iv. "**Document**" means the information contained in any electronic file, which has been published by the Licensor's as AXMEDIS official document and listed in the web site mentioned above or available by any other means.
   - v. "**Works**" means any works created by the licensee, which reproduce a Document or any of its part.
2. **LICENCE**
   1. The Licensor grants a non-exclusive royalty free licence to reproduce and use the Documents subject to present terms and conditions (the **Licence**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.
   2. In consideration of the Licensor granting the Licence, licensee agrees to adhere to the following terms and conditions.
3. **TERM AND TERMINATION**
   1. Granted Licence shall commence on Acceptance Date.
   2. Granted Licence will terminate automatically if licensee fails to comply with any of the terms and conditions of this Licence.
   3. Termination of this Licence does not affect either party's accrued rights and obligations as at the date of termination.
   4. Upon termination of this Licence for whatever reason, licensee shall cease to make any use of the accessed Copyright.
   5. All provisions of this Licence, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this Licence and shall continue in full force and effect.
   6. Notwithstanding License termination, confidentiality clauses related to any content, document or part of it as stated in the document itself will remain in force for a period of 5 years after license issue date or the period stated in the document whichever is the longer.
4. **USE**
   1. Licensee shall not breach or denigrate the integrity of the Copyright Notice and in particular shall not:
      - i. remove this Copyright Notice on a Document or any of its reproduction in any form in which those may be achieved;
      - ii. change or remove the title of a Document;
      - iii. use all or any part of a Document as part of a specification or standard not emanating from the Licensor without the prior written consent of the Licensor; or
      - iv. do or permit others to do any act or omission in relation to a Document which is contrary to the rights and obligations as stated in the present license and agreed with the Licensor
5. **COPYRIGHT NOTICES**
   1. All Works shall bear a clear notice asserting the Licensor's Copyright. The notice shall use the wording employed by the Licensor in its own copyright notice unless the Licensor otherwise instructs licensees.
6. **WARRANTY**
   1. The Licensor warrants the licensee that the present licence is issued on the basis of full Copyright ownership or re-licensing agreements granting the Licensor full licensing and enforcement power.

2. For the avoidance of doubt the licensee should be aware that although the Copyright in the documents is given under warranty this warranty does not extend to the content of any document which may contain references or specifications or technologies that are covered by patents (also of third parties) or that refer to other standards. AXMEDIS is not responsible and does not guarantee that the information contained in the document is fully proprietary of AXMEDIS consortium and/or partners.
3. Licensee hereby undertakes to the Licensor that he will, without prejudice to any other right of action which the Licensor may have, at all times keep the Licensor fully and effectively indemnified against all and any liability (which liability shall include, without limitation, all losses, costs, claims, expenses, demands, actions, damages, legal and other professional fees and expenses on a full indemnity basis) which the Licensor may suffer or incur as a result of, or by reason of, any breach or non-fulfilment of any of his obligations in respect of this Licence.

7. INFRINGEMENT

1. Licensee undertakes to notify promptly the Licensor of any threatened or actual infringement of the Copyright which comes to licensee notice and shall, at the Licensor's request and expense, do all such things as are reasonably necessary to defend and enforce the Licensor's rights in the Copyright.

8. GOVERNING LAW AND JURISDICTION

1. This Licence shall be subject to, and construed and interpreted in accordance with Italian law.
2. The parties irrevocably submit to the exclusive jurisdiction of the Italian Courts.

Please note that:

- You can become affiliated with AXMEDIS. This will give you the access to a huge amount of knowledge, information and source code related to the AXMEDIS Framework. If you are interested please contact P. Nesi at nesi@dsi.unifi.it. Once affiliated with AXMEDIS you will have the possibility of using the AXMEDIS specification and technology for your business.
- You can contribute to the improvement of AXMEDIS documents and specification by sending the contribution to P. Nesi at nesi@dsi.unifi.it
- You can attend AXMEDIS meetings that are open to public, for additional information see WWW.AXMEDIS.org or contact P. Nesi at nesi@dsi.unifi.it

1 Executive Summary and Report Scope

Market and end-users are pressing content industry to reduce prices. This is presently the only solution to setup viable and sustainable business activities with e-content. Production costs have to be drastically reduced while maintaining product quality. Content providers, aggregators and distributors need innovative instruments to increase efficiency. A solution is automating, accelerating and restructuring the production process to make it faster and cheaper. The goals will be reached by: (i) accelerating and reducing costs for content production with artificial intelligence algorithms for content composition, formatting and workflow, (ii) reducing distribution and aggregation costs, increasing accessibility, with a P2P platform at B2B level integrating content management systems and workflows, (iii) providing algorithms and tools for innovative and flexible Digital Rights Management, exploiting MPEG-21 and overcoming its limits, supporting several business and transactions models. AXMEDIS consortium (producers, aggregators, distributors and researcher) will create the AXMEDIS framework with innovative methods and tools to speed up and optimise content production and distribution, for *production-on-demand*. The content model and manipulation will exploit and expand MPEG-4, MPEG-7 and MPEG-21 and others real and de-facto standards. AXMEDIS will realise demonstrators, validated by means of real activities with end-user by leading distributor partners: (i) tools for content production and B2B distribution; (ii) content production and distribution for i-TV-PC, PC, kiosks, mobiles, PDAs. The most relevant result will be to transform the demonstrators into sustainable business models for products and services during the last project year. Additional demonstrators will be 2-3 associated projects launched as take up actions. The project will be supported by activities of training, management, assessment and evaluation, dissemination and demonstration at conference and fairs.

This deliverable is devoted to the description of content protection and supervision done inside WP4.5.

This activity is by no means finished with the completion of this deliverable, but it has to be revised during the development of the project.

Main deliverables in WP4 are:

- DE4.1.1 – Content Modelling and managing (M13), report and prototype;
- DE4.2.1 – Content indexing, monitoring and querying (M13), report and prototype;
- DE4.3.1 – Content Composition and formatting (M13), report and prototype. It also includes the details about the integration of AXMEDIS with workflow management tools;
- DE4.4.1 – Content sharing and production on P2P (M13), report and prototype;
- DE4.5.1 – Content Protection and Supervision (M13), report and prototype;
- DE4.6.1 – Content Distribution via Internet (M13), report and prototype;
- DE4.7.1 – Content Distribution toward mobiles (M13), report and prototype;
- DE4.8.1 – Content Distribution via satellite data broadcast, the push optimisation and the on demand problem (M13), report and prototype;
- DE4.9.1 – The Usability issues for the AXMEDIS production tools (M13), report.

The main activities that have supported the production of this deliverable are related to:

WP4.2 - Content Indexing, Monitoring and Querying – In this sub-workpackage, the main functional aspects for accessing content are specified, designed and described. For the specification existing standards have been considered.

- The Crawling and indexing collector engine fulfils the requirements of collecting content from existing databases. This allows users to easily transfer existing content stored in their content management systems to be transferred into the AXMEDIS framework.

- The definition of content descriptors considers the standardisation activities for meta data descriptos in MPEG-7. Thus, the content descriptors used within AXMEDIS are compatible and can be exchanged with other solutions that are able to read MPEG-7 meta data descriptors.

- Like the general content descriptors, fingerprints are compatible with the MPEG-7 descriptors. Additionally the requirements of MPEG-21 are consider for content authentication and verification as well as for content identification within DRM environments.

- Query Support considers existing standards for identifying and accessing content.

- In Query Support for PCs a prototype for accessing content and digital item adaptation based on a general client profile is developed.

2 Introduction

The integrated, flexible and scalable AXMEDIS solution addressing content creation independent of the content type and multi-channel content distribution faces this challenge: providing access to content by effective content indexing and querying techniques as well as monitoring the content distribution and content exchange in a scalable way.

These content indexing, monitoring and querying techniques are essential for the content creation as well as for content consumption. Thus, the main goals of the research and development activities described in this deliverable address these fundamental parts of the AXMEDIS framework.

2.1 Specification of T4.2.1 Crawling and Indexing, Collector Engine

Major partners involved

Managed by DSI and realized with a subcontract already identified in the contract.

Research and development plan

The aim of this task is indexing of the databases. This is achieved crawling into the CMSs and the databases of the content providers, integrators, and distributors to access content to indexing it. Specific interfaces for accessing to these databases will be defined and developed in WP9. The general infrastructure and the integration of content indexed with the AXMEDIS data model and database identified in WP4.1 have to be produced in this WP.

- Corresponding to the requirements of the project, the subWP leader will design and develop a framework which allows collecting and indexing different media sources (databases, file systems, etc.)
- The idea is to access already available information and metadata into CMSs and transfer it into a cache of the Content (see also WP4.1). Those may have information related to the content, their metadata, their technical details, DRM aspects, licensing aspects, possible products that use the content, multilingual information, etc.
- Tool for collecting content from several sources. Indexing content contained into the Content management systems of the content distributors and producers. This tool is called in the General Architecture as **Crawler Collector Indexer**. It may navigate on any kind of source by means of special plug-ins that will be developed in the subWP related to the content distributors in WP4 and in the subWPs of WP9
- This task also includes the development of a tool called **Collector/Transcoder Engine** that will allow accepting the maps defined in the WP4.1 to migrate the content from the Crawled Results Integrated Database to the AXMEDIS format and database. These tools will be capable of processing data automatically updating the content into the AXMEDIS database when these are updated into the CMSs. In addition, the same engine tool will be capable to integrate tools for direct estimation of fingerprint to complete the indexing by using the results of the following task
- The Collector Engine will be endowed of a specific interface to receive from the query support queries for searching into the Crawler Results Integrated database and sending back the information to the AXMEDIS database manager

This task will be performed by acquiring the license of a product called Focuseek and integrating it in the AXMEDIS framework. The license will come with editable manual to customize the installation to the AXMEDIS solution. Moreover Focuseek will also participate for the development of an interface with its product (see T4.2.5).

The work to be done has been described in terms of requirements into the requirements document, while the specification of integration is reported into the AXMEDIS framework specification.
The major components that will be studied and realized in this Task are:

- Crawler Collector Indexer, commercial product
- Crawler Results Integrated Database, commercial product
- Collector Internal Engine, to be customized according to our needs
- Collector Plug in Manager, commercial product to be marginally adapted
- Metadata mapper, to be customized according to our needs (this module is used for:
    - Mapping metadata information from the Crawler Results Integrated Database to the AXMEDIS Object Model
    - Mapping query coming from the AXMEDIS world to the Crawled Database
    - Mapping results of the query coming from the AXMEDIS world to the Crawled Database

Planned schedule

- M6: signature of the contract with the company that provide the above components and customizations
- M7: full documentation of the above mentioned commercial product
- M8: detailed design of Collector Internal Engine and of the Metadata mapper
- M10: delivering of the tools and start of the trial of them on some content provider
- M18: installation to several content providers and collection of data, setting of the active queries, etc. Usage and test of the first version of the Crawler and Indexing. Content acquisition with customizable engine for navigating in sources and collecting content and indexing it
- M18: entering in a phase of pure maintenance and update according to the contract

2.2 Specification of T4.2.2 Estimating Fingerprint for Indexing

Major partners involved

FHGIGD, EPFL, DIPITA

State of the Art

This task is rather complex and involves several signal processing disciplines; it includes research for fingerprinting in relation to Task 4.5.4 (see later in this document) as well as research for the extraction of meaningful metadata in relationship with T4.3.1 (see next section). For metadata, a special emphasis in research will be on music, sound and speech, while video analysis, a rather complex domain indeed, will be mainly integrated in the AXMEDIS Framework by means of existing tools and technologies.

For fingerprinting technologies different research prototypes have been developed. Some of them have grained to products. Due to the requirement of perceptual similarity the state of the art is quite complex when the broad range of different technologies is considered. Their development aimed at the identification of content. The research and development activity considers existing state of the art (e.g. MOUMIR, Models for Unified Multimedia Information Retrieval: http://www.moumir.org/). Different free and commercial libraries are available for the implementation, which will be the basis of the research activities within AXMEDIS (e.g. extensive lists are available for audio: http://music-ir.org/evaluation/tools.html and for images and video: http://www-2.cs.cmu.edu/~cil/v-source.html).

Research on high-level descriptors of musical sounds and musical pieces will be based on state of the art research, including the Cuidado (http://ismir2002.ismir.net/proceedings/02-FP06-3.pdf) and the Semantic Hi-Fi (http://shf.ircam.fr/) Europeans projects. In particular the free CLAM framework already used in the Cuidado project (http://www.iua.upf.es/mtg/clam/) is of interest for this research domain. The needed machine learning algorithms will be taken from free libraries such as Torch (http://www.torch.ch/), Weka (http://www.cs.waikato.ac.nz/ml/weka/) or LibSvm (http://www.csie.ntu.edu.tw/~cjlin/libsvm/).

For the extraction of descriptors regarding speech, present technologies for marking voice/music transitions and inter-speakers transitions are considered and implemented like the VILE project

(http://liceu.uab.es/~joaquim/ phonetics/VILE/VILE_SEAF03_Engl.pdf). Research for the extraction of high level descriptors of text files must consider a multilingual environment. Within a multilingual framework, the main available tools for PoS-tagging are
(a) the Connexor Machinese Phrase Tagger, based on linguistic modeling, (http://www.conexor.fi/),
(b) the Xerox Finite State part-of-speech tagger (http://www.xrce.xerox.com/competencies/content-analysis/fsnlp/tagger.en.html), based on HMM statistical modeling, (c) the TreeTagger (http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html#Linux), another statistical one.
Frequency lexicons are also needed as referring lexical data bases for the automatic detection of domain specific terminology in a document (see Drouin, 2004; the BNC word lists and frequencies are available online at http://www.itri.bton.ac.uk/~Adam.Kilgarriff/bnc-readme.html#lemmatised)
Keywords extraction and summarization processing (that rely on the precedent ones) can be mainly based on two main models: (1) the use of statistical methods (see Nguyen http://www.jaist.ac.jp/library/thesis/is-doctor-2005/paper/nguyenml/paper.pdf) , (2) the use of ontologies/thresauruses/dictionaries/frequency lists as basis for lexical and semantic analysis (see Bellare et al. 2004, about the use of WordNet for the summarization task; WordNet site at http://www.cogsci.princeton.edu/~wn/).
Review of existing libraries for textual documents format conversion is also needed, e.g. Html2text (http://userpage.fu-berlin.de/~mbayer/tools/html2text.html),
ps2ascii (**Errore. Riferimento a collegamento ipertestuale non valido.**).

Research and development plan

Research in this task is dedicated to the processing of content in its several forms and formats and producing specific technical metadata and high level information that can be used for fingerprinting and for content search and retrieval. The creation of the information necessary for retrieval can be performed when the content is processed to be moved and stored into the AXMEDIS database (*Content Indexing*). The feature extraction functionality is also used in the distribution tool for monitoring the content distribution (*Content Monitoring*). The feature extraction and calculation depends on the application (indexing or monitoring). As said, this Task will be developed in conjunction with work done in Task 4.5.4; in that task focus will mainly be on purely fingerprinting research aiming at content protection, whereas in this task all technologies concerning the feature extraction and indexing will be joint in a unique framework.

Research and development within the first months will initially be based on the state of the art and its integration within the AXMEDIS framework. After that, steps towards innovations will follow. In general, the following steps can be identified:

1. Identification of the research prototypes and products available for the integration within the AXMEDIS framework
2. A pre-selection will be based on this first analysis. This analysis considers the publicly available information like scientific publications, technical reports or white papers
3. The pre-selected prototypes or products will be evaluated in practical tests. For these tests the specific requirements within AXMEDIS will be the evaluation foundation
4. Identified weaknesses or drawbacks will initiate the improvement of existing methods and will lead to the development of new methods

For what concerns descriptors, low level descriptors for the media types exchanged within AXMEDIS will be investigated, but they will mainly constitute a starting point for higher level analysis, much more relevant for the AXMEDIS automation of content creation procedures.

For high level descriptors extraction for audio and musical pieces potential issues and criteria for the analysis, development, and improvement include:

- General descriptors (e.g. based on CLAM and other state of the art), including perceived energy, global timbre color, instrument recognition, rhythmic descriptors and rhythmic patterns, melodic descriptors, harmonic descriptors, etc.

- Formalization of descriptors in terms of MPEG-7 or MPEG-21
- Automatic detection of the musical genre of an audio piece: e.g. definition of taxonomy of musical genre, use of high level descriptors and machine learning algorithms to compare an audio piece to the pre-selected musical genres
- Automatic generation of music audio summaries, based first on the segmentation and determination of the musical structure of the piece and then automatic selection and combination of the most representative part of the piece

Test databases will include among others materials provided by ISMIR conference for its Audio Description Contest (http://ismir2004.ismir.net/ISMIR_Contest.html) and the RWC Music Database (http://staff.aist.go.jp/m.goto/RWC-MDB/)

For high level descriptor extraction for audio reporting speech potential issues and criteria for the analysis, development, and improvement include:

- Specification of features characterizing turns and voice alternation and the corresponding length of dialogic turns.
- Inference of speech types (dialogue, monologue)
- Feature extraction and generation of metadata of a audio tracks through automatic detection of music - voice alternation and generation of the corresponding lengths

For high level descriptor extraction for contents comprising texts potential issues and criteria for the analysis, development, and improvement include:

- Evaluation of performance of various multilingual enabling technologies (supporting at least English, German, French, Spanish, Italian)
- Acquisition of frequency
- Identification of language by means of reference dictionaries
- PoS tagging
- Keywords extraction by means of comparison with reference frequency lexicons

High level descriptors extraction for video objects is focused on the state of the art development. State-of-the-art algorithms will be analyzed, including:

- GIFT (Gnu Image Finding Tool), http://www.gnu.org/software/gift/
- Viper, University of Geneva, http://viper.unige.ch/
- and other solutions considered by TREC Video Retrieval Evaluation, http://www-nlpir.nist.gov/projects/trecvid/

Relevant solutions will be identified. Potentially relevant descriptors are content descriptors

- color (histograms)
- motion (kinetic energy, Mpeg motion, optical flow)
- edge information
- texture information
- text elements (video-ocr)

A general interface will be provided for the integration of existing solutions and selected existing solutions will be integrated.

In general, the descriptors will be formalized when possible in terms of MPEG-7. In addition, the content for validation of the descriptors and of their estimator algorithms will be tested against reference test cases similar to those of MPEG and including those of MPEG-4 and -7. In any case an XML loader and saver engine for descriptors in XML will be used.

The feature extraction has to be performed by implementing algorithms for descriptors estimation and defining specific MPEG-7 or MPEG-21 descriptors.

Planned schedule

- M8 Test database definition and collection
- M11 Evaluation of state of the art technologies for low and high level content descriptors
- M12 Identification of critical issues in the association of content descriptor extraction. Identification of the first versions of the content descriptors algorithms. Integration of available algorithms and evaluation based on the test cases.
- M14 First version of the set of low-level content descriptors algorithm for video, audio, and text files (in terms of MPEG-7 or MPEG-21 if applicable). In relation to T4.5.4
- M14 First version of the set of high level descriptors algorithm for audio files (in terms of MPEG-7 if applicable)
- M18 An experimental fingerprint tool for extracting low and high level content descriptors. Content is processed offline. Selection of the best algorithms, integration of these solutions and their improvement. In relation to T4.5.4
- M18 Revised first version of the set of fingerprint estimation algorithm for video, audio, and text files. In relation to T4.5.4
- M18 First version of the musical genre recognition algorithm and of the audio summary generation algorithm

Detailed schedule for sound and music

- M8 Test database definition and collection
- M11 Evaluation of state of the art technologies for low level descriptors for audio and music (audio processing libraries)
- M12 Evaluation of state of the art technologies and resources for high level descriptors for sound and music.
- M12 First version of new algorithms in fields where suitable technologies are not available
- M18 Integration of fingerprint and descriptors extractors in fingerprint tool and descriptor extractor tool

Detailed schedule for speech and voice

- M8 Testing corpora collection
- M11 Evaluation of state of the art technologies for low level descriptors for text and audio (file formats converters, text and audio processing libraries). Evaluation of state of the art technologies and resources for high level descriptors for text (multilingual dictionaries, PoS taggers, document fingerprinting algorithms).
- M12 First version of new algorithms in fields where suitable available technologies were not found
- M14 First version of low and high level descriptors and their conversion in MPEG-7 format
- M18 Integration of fingerprint and descriptors extractors in fingerprint tool and descriptor extractor tool

2.3 Specification of T4.2.3 General Query Support in the AXMEDIS database

Major partners involved

EXITECH

State of the art

The query support is a combination of query mechanisms for retrieving content objects among the indexed information, in the AXMEDIS database (for content processing, composition, formatting, etc.), on the P2P AXEPTool, on the content collector referencing the content contained in the CMSs (Crawled Results Integrated Database). The present technology to formalize the queries will be reviewed and the most flexible and diffuse model and format will be chosen for the formal definition of queries. Architectures of the queries will be analyzed and mapped on the AXMEDIS need to assess them.

Research and development plan

This Task is about the definition of a common format for:

- Sending queries and exchanging results among AXMEDIS database, AXEPTool peers, Crawled Results Integrated database, and the internal structure of the AXMEDIS objects, problems of mapping or metadata, problems of mapping of different languages, etc.
- Accepting queries requested from other tools such as Formatting Engine, Composition Engine, etc.
- Studying a specific user interface integrated into with the other tools defining technical queries and Selections and it is integration into WP4.4; Balancing complexities of the structure of the queries with the real need of the users and the complexity of the metadata.
- Studying performance of the database in the case of retrieval and different indexing models and their impact in the general model for all databases (M30)

Planned schedule

- M13: First version of query Support for AXMEDIS database including query distribution and integration in draft;
- M14: first version of the mapper for processing queries and collecting results on the crawled information
- M24: Final version of the service interface to allow the expression of queries and Selections from other tools and the reception of results

2.4 Specification of T4.2.4: The query support in the AXMEDIS database

Major partners involved

EXITECH

State of the art

This task is developed moving from the analysis and model specified in T4.2.3; specialized query tools for media searching for the distributors and aggregators will be developed. This part is performed in collaboration with T4.4.1 of WP4.4 regarding query for the distributors. The query search engine will be joined in the B2B-P2P model for sharing of content components among producers and distributors.

According to analysis of the state of the art and requirements, the query engine will be powerful and will provide the usage of:

- details related to the description of components, see below for the formatting
- costs and DRM rules, for each action a price, play, excerpts, redistribution, resizing, distribution on a different area, validity of the DRM rule and copyright coverage, etc.
- available languages if there is speaker or text
- range of age suggested
- business model suggested
- time of delivering and availability in terms of first delivering, if not ready
- type of delivering: on-line, offline, etc.
- if on-line time of downloads or acquisition
- cultural level
- subject, description of content with simple metadata
- textual description of subject and evolution

Research and development plan

This task is about the definition of a common format for:

- The specification of technical/professional query including metadata, technical information, business and licensing aspects, content based, DRM rules, etc. The query has to be based on the AXInfo data

structure. In addition, other metadata can be added dynamically by the users according to the AXMEDIS Model These metadata has to be managed dynamically in order to avoid restriction in using additional metadata
- Balancing the performance and the flexibility of the database structure considering the needs of openness in terms of metadata and the needs of performance to give answers in reasonable time on very large databases
- Studying performance of the database in the case of retrieval and different indexing models in the AXMEDIS database (M30)

Planned schedule

- M11: Analysis of the query support, definition of the first version of support model for technical queries;
- M12: first version of the service interface to allow the expression of queries and Selections from other tools and the reception of results
- M24: First version of the specialized query tools for multimedia media searching for the distributors, aggregators and end users;
- M30: Final version of the specialized query tools for multimedia media searching for the distributors, aggregators and end users;

2.5 Specification of T4.2.5: Collector Engine and query support in the database of Crawler Results Integrated database

Major partners involved

Managed by DSI and realized with a subcontract already identified in the contract.
More specific details will be provided to the subcontractor to better specify its work and integration with AXMEDIS.

Research and development plan

This task includes the development of a tool called Collector/Transcoder (recently called Mapper) Engine that will allow accepting the maps defined in the WP4.1 to migrate the content from the Crawled Results Integrated Database to the AXMEDIS format and database. These tools will be capable of processing data automatically updating the content into the AXMEDIS database when these are updated into the CMSs. In addition, the same engine tool will be capable to integrate tools for direct estimation of fingerprint and possibly metadata to complete the indexing by using the results of the following task, T4.2.6.

The Collector Engine will be endowed of a specific interface to receive from the query support queries for searching into the Crawler Results Integrated database and sending back the information to the AXMEDIS database manager.

For query support, it will be developed starting from the analysis and specified model in T4.2.3. The above query for the AXMEDIS database has to be also forwarded (when explicitly requested) into the database of information and content of the CMS, which is present in the content factory (content provider, content integrator, etc.). The information obtained as a result has to be integrated with the results obtained from the of the AXMEDIS database general query. The results of the query will not have to include equivalent content that is already available on the AXMEDIS database since it is automatically pushed in by the Collector Engine tool.

Planned schedule

This Task share the development plan of Task T4.2.1
- M6: signature of the contract with the company that provide the above components and customizations
- M13: First version of query support for Crawler Content integrated with that of the AXMEDIS database

- M18: Final version of query support for Crawler Content integrated with that of the AXMEDIS database;
- M18: entering in a phase of pure maintenance and update according to the contract

2.6 Specification of T4.2.6: Query support for connecting AXMEDIS tools with content distributors

Major partners involved

FHGIGD

State of the art

The goal of this task is it to connect the AXMEDIS query support with content distributors to provide an interface to pass queries from the final clients on into the AXMEDIS framework. Therefore queries performed by final users (done in XML, SQL or OSQL) on the distributor server have to be traduced for the AXMEDIS database. The query results have to be traduced back for the distributor server and thus for the final clients. The present technology to formalize and to traduce queries as well as existing query tools for media search will be reviewed and the most flexible and appropriate models and formats will be chosen to connect the AXMEDIS tools with the content distributors. Existing architectures and technologies will be analyzed and mapped to the requirements within the AXMEDIS framework to assess them.

Research and development plan

This task is mainly focused on establishing the possibility for the distributors to pass into the AXMEDIS tools and database the queries performed by the final clients on their devices, and return back the corresponding results (managing the production of content via the Formatting Tools). The final clients are interested in making queries to the content distributor server to get content over that has been explicitly promoted by Programme, etc. (content on demand). Thus the user could ask to have content on demand out of the proposed set of content for the week or for the month, which is in general over the content programmed for the distribution. This content can be ready to be distributed and thus available for distribution or could be created/formatted on demand.

Technically, as mentioned before the query performed by the final users (done in XML, SQL or OSQL) on the distributor server has to be traduced for the AXMEDIS database passing it to the Programme and Publication tool engine. The results are produced in some way and have to be traduced back for the distributor server and thus for the final clients. Other related activities are specialized query tools for media searching for the end users.

The results produced will be used in WP9.5.1 and WP9.6.1 and related subWPs. In those cases, the information needed to identify the single object into a database is more related to the content description and metadata rather than on technical details as in the previous one.

Planned schedule

- M18: First draft of the specialized query tools for media searching for the distributors, aggregators and end users
- M24: First version of the specialized query tools for media searching for the distributors, aggregators and end users
- M36: Final version of specialized query tools for media searching for the distributors, aggregators and end users

3 Content Flow within AXMEDIS

As described in [Bazea-Yates99] information retrieval addresses different areas:

- Representation of information items

- Information item storage
- Organisation of information items
- Access to information items

These aspects are addressed within AXMEDIS in the corresponding work packages:

- Content within AXMEDIS are crawled and indexed. This is done by the collector engine (see also the document DE3.1.2 – Part E – database gathering).

- For each content, content descriptors are automatically calculated. Within AXMEDIS low level and high level descriptors, as well as fingerprints are considered. So far, the development focus within AXMEDIS is on audio, video and text descriptors. Fingerprinting methods are developed for text, audio, image and video content (see also the document DE3.1.2 – Part D – fingerprinting descriptors).

- These content descriptors as well as other stored information are accessible via the so-called query support. Additionally, an interface is provided for clients. The demonstrator is query support for PC using web interfaces (see also the document DE3.1.2 – Part E – database gathering).

In the following chapters the state of the art, the problems identified, and the work performed for each of the relevant tasks are described.

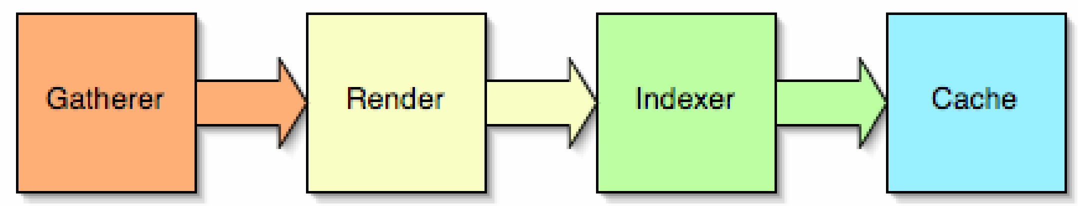
4 Crawling and Indexing, Collector Engine

4.1 Crawling and Indexing Process

A very important aspect of the AXMEDIS project is about the interfacing with data repository of Content Providers. The Content Providers world is very heterogeneous so that the integration process with AXMEDIS system should be mediated by a flexible gathering system. The goal of this approach is to minimize the impact of AXMEDIS system on the infrastructure of Content Providers that must continue using their old archiving and publishing system without any modification.

focuseek is a multimedia Content Gathering and Indexing platform whose main goal to manage huge amount of data coming from different and distributed information sources. focuseek architecture can be successfully used to implement Information Retrieval projects for large enterprises, government institution and large internet portals.

focuseek platform can collect information from various kind of sources, implement a way to analyze the gathered content and provides a very flexible, high performance indexing system for content retrieval.



In the above figure, main components of focuseek system are shown.

The ***Gatherer*** is the coordinator of a pool of gathering agents whose task is to acquire new data from an information source, as soon it is available. A specific implementation of a gathering agent is for instance a Web Crawler that, starting form a set of initial seed Web pages is able to explore a portion of the Web using a breadth first strategy to visit its graph. focuseek substitute the concept of “web crawler” with the more flexible concept of "Source+Gatherer". With this metaphor, focuseek is able to natively acquire information from a various type of remote sources like Databases, Web Services, News Servers, WebDav, IMAP folders, Filing Systems, and some other proprietary sources. The Gatherer module is fully programmable and fully customizable writing Plug-Ins for specific sources.

The **Renderer** is a very central component in the focuseek architecture. focuseek indexing and retrieval system does not work on the original version of data but on a so called "rendered version". The idea is that from any piece of information (a document, for instance) a set of features can be extracted if we provide a suitable algorithm for that. A typical example is a portion of text where the extracted features are the list of words contained in it or a bitmap image from which some text can be extracted by an OCR engine. Obviously more sophisticated extraction algorithm must be used for different type of information (like images of sounds) but the main concept does not change. All extracted features are then compiled in an internal XML format and passed to the indexing module. The extraction process of renderer component is executed by a pipeline of Plug-Ins, which provide the compilation of the internal XML representation. focuseek provides some default Plug-Ins for some basic type of content and an API to write custom Plug-Ins.

The **Indexer** creates the index of a collection of information gathered from multiple sources and offers a complete query language for retrieving original contents. The index produced is fully dynamic in the sense that any indexed content is almost immediately available for the query. This is a crucial feature for all always-on systems because there is no need of periodically index reconstruction.

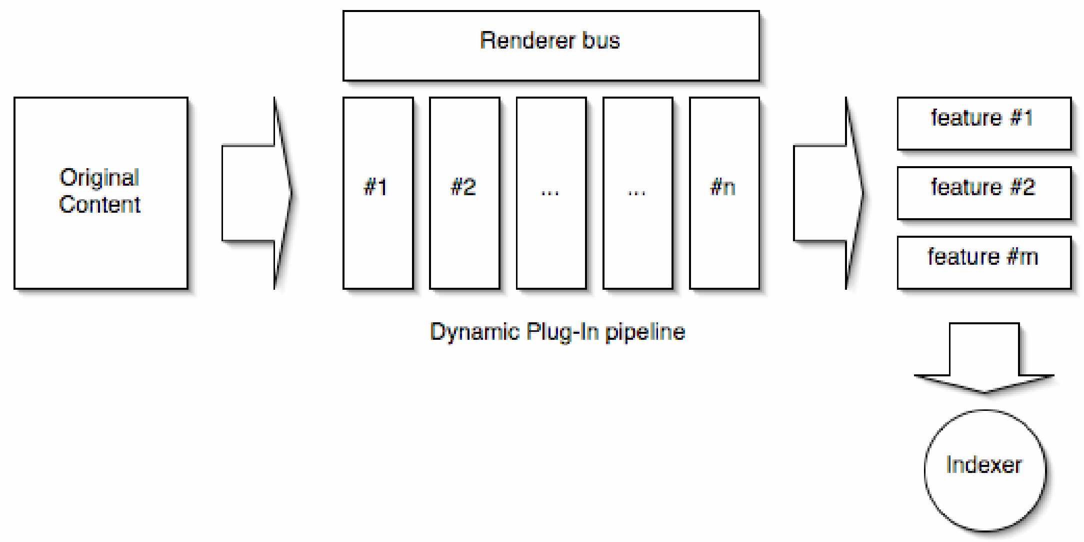
The content **Cache** is highly coupled with the index and acts like a cache for the content that needs to be locally mirrored. It is a multilevel cache and it can be used to store, and index, multiple version of the same content. Setting depth parameter equal to zero, no content will be mirrored.

Some main technical objectives had been followed during *focuseek* development:

Minimum environmental impact
Information access in complex organizations, with multiple data sources, multiple document formats, multiple access permission, etc. is often a very difficult task to manage. focuseek gathering approach is very straightforward and assumes as default the worst case: the data source is reachable but it is like a black box. It means that a gathering agent can access to data of a source using its standard access interface but it cannot influence the way of data are provided or formatted into the source itself. In addition, the agent is not interested in the internal format of such data because it has only to cope with its public interface. To be successful in this kind of approach the focuseek gathering system is highly adaptable to various source and data format, in many cases it is able to automatically obtain needed information from a source sniffing, in other cases it needs a manual configuration; in the worst case a gathering plug-in must be written.

Flexibility
The same flexibility of the gathering system is exploited by the indexing system too. The focuseek indexer module can manage any feature that a specific parsing module (see Renderer) is able to extract from original raw content. All extracted and indexed feature can be combined in the query language exposed by the query interface of the indexer module.

In the following figure, we can see the original content that is filtered by a Plug-In pipeline where each Plug-In is specialized in extracting a specific feature. Extracted features are then transformed in metadata and passed to the indexer module.



As example, in the case of musical content, a pipeline of Plug-In able to extract information like music class, type of instruments, etc. will provide to the indexer a set of useful selective features that can be used in the retrieval process directly from the query interface.

As default, focuseek provides Plug-Ins to extract text from a various types of documents: HTML, XML, TXT, PDF, PS, DOC, RTF. Other formats can be added with specific Plug-Ins.

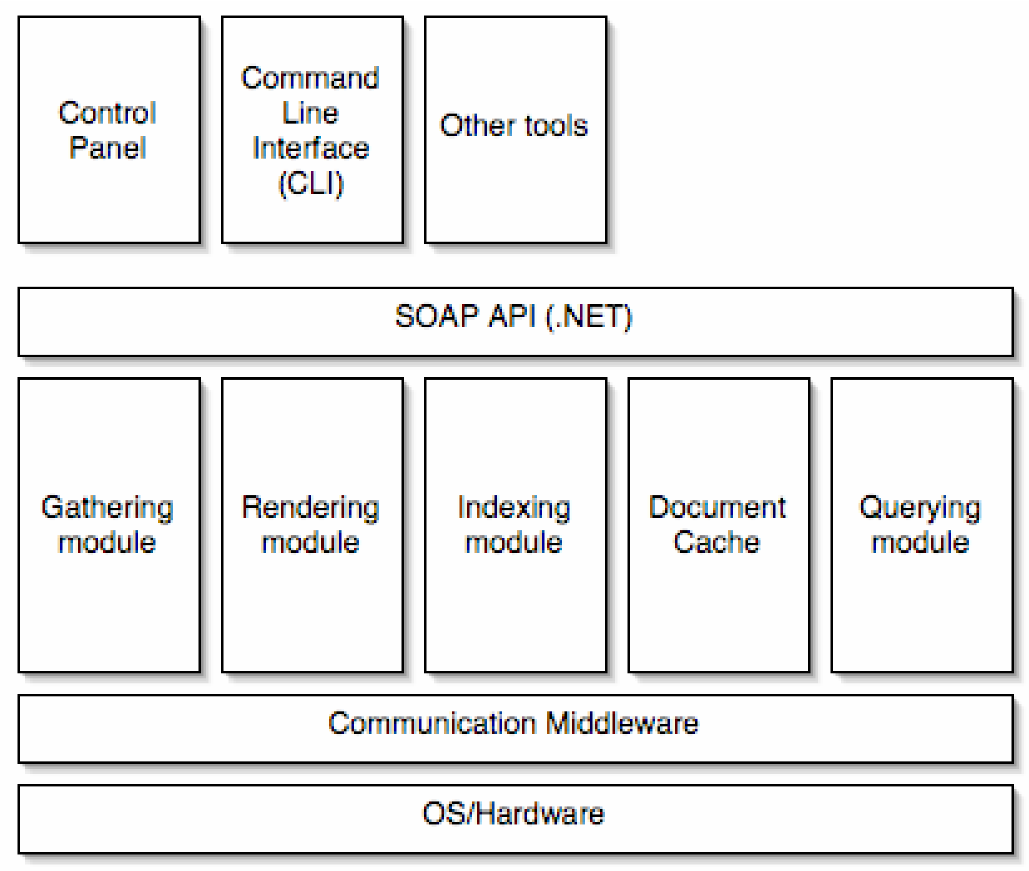
Performance
Every component of focuseek system is implemented with a custom technology focused to obtain very high performance in every situation regarding retrieval of information from unstructured repositories. In such sense, critical parameters are the gathering and indexing speed. In a current version of the system a single server, single processor installation is able to index up to one million of documents (a document is conventionally defined as a content with 1000 index-able attributes. For a text document, those attributes are, for instance, the words composing it), it supports more than 20 query/sec and can reach a gathering transfer rate of about 10 docs/sec depending from the available bandwidth.

Dynamism
focuseek is a native full dynamic system. It has been developed with the main goal to manage very dynamic information so that all changes in any of the monitored source is almost immediately propagated in every component of the system with a latency time of few seconds. focuseek has been though as a 24/24h, 7/7d online system so it does not need periodical index rebuilding and all other maintenance work can be made online directly on the production environment.

Scalability
focuseek supports both vertical and horizontal scalability. Due to its component based architecture and its fully multithread implementation, focuseek is able to take full advantage of multiprocessor server deployment. For huge implementations, a deployment model based on GRID approach is also possible. In this case, all focuseek components are deployed on a suitable number of single processor, low cost server connected together in a local or geographical network. A communication infrastructure between nodes is provided by focuseek itself.

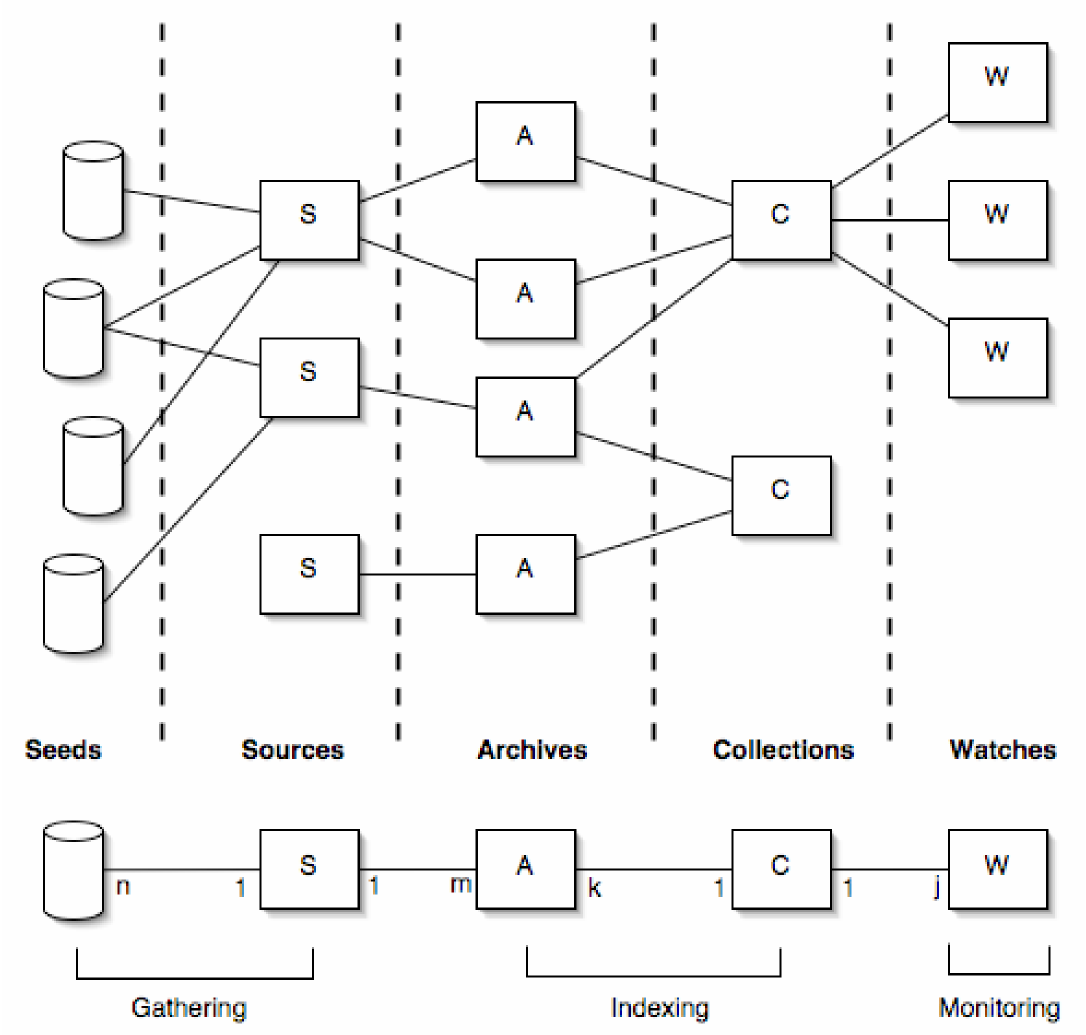
4.1.1 The focuseek architecture

The following figure shows the physical components of current focuseek architecture



focuseek is a component based platform completely developed in C++ language and available for Windows, Linux and OSX operating systems. All focuseek features are accessible using a complete SOAP API that is compliant with the latest Microsoft .NET standard. A complete suite of administrative tools comes with focuseek both as GUI and Command Line.

This other figure shows how the main concepts exposed by focuseek are connected to each others.



From the user point of view focuseek offer some basic concept that can be combined together in order to achieve the needed gathering and indexing functionalities. Such concepts are divided in three groups according with the SOAP API:

Gathering group. The central gathering concept is the Source. The goal of a Source is grouping a certain number of Seeds and configuring for them a suitable access protocol. A seed can be a database table or a Web Page, a complete Web Site, a specific portion of the Web or a fully custom data repository. The Source natively supports access to seeds with digital certificate, password, cookies etc. A Seed can be shared by many Sources.
The current version of focuseek platform is natively able to gather information from the following types of sources:

Breadth-First Web Crawling	from a set of initial seeds a web crawler can explore the Web Graph recursively following hyperlink contained in.
Site Crawling	same of above but the crawling activity is limited to the sites containing original seeds.
File Systems	any local or remote file system, accessible from the server where focuseek platform is installed

Newsgroups	usenet groups through NNTP protocol
DB	direct access to database tables. Compatibility with Oracle, DB2 and MySQL. An ODBC adapter is also available.
IMAP4	any mail folder accessible through an imap server.
POP3	same of above but with POP access protocol
WebDAV	a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers
RSS feeds	a format for syndicating news and the content of news-like sites

The gathering module provides an application interface (API) that can be used to implement new gathering protocols in order to match custom data repositories.

Indexing group. An Archive represents an index of contents coming from a Source. Multiple Archives can be connected to a single Source because every Archive can have different configuration rules to create the index (i.e. caching or no caching). In order to create indexes from different kind of Source many Archives can be grouped together by a Collection. Both Archive and Collection are query-able objects.

Monitoring group. focuseek can be used as monitoring tool too. Watches are contains a set of static filters on the information stream coming from a Collection so that they can be used to implement a custom view on any information stream. Watches supports subscriptions from any client application that needs to be alerted as soon a specific condition is matched.

4.2 State of the Art

The following is a brief summary of the most important products dealing with content crawling and indexing:

Google Appliance (www.google.com/appliance/)
It is the hardware/software solution from the most famous search engine company of the world. The appliance implements exactly the product available on the web but with a specific focus on the large enterprise market.

Verity Ultraseek (www.verity.com/products/ultraseek/)
Verity had recently acquired Inktomi search engine software division. Ultraseek is a pure search engine solution very well integrated with other Verity products.

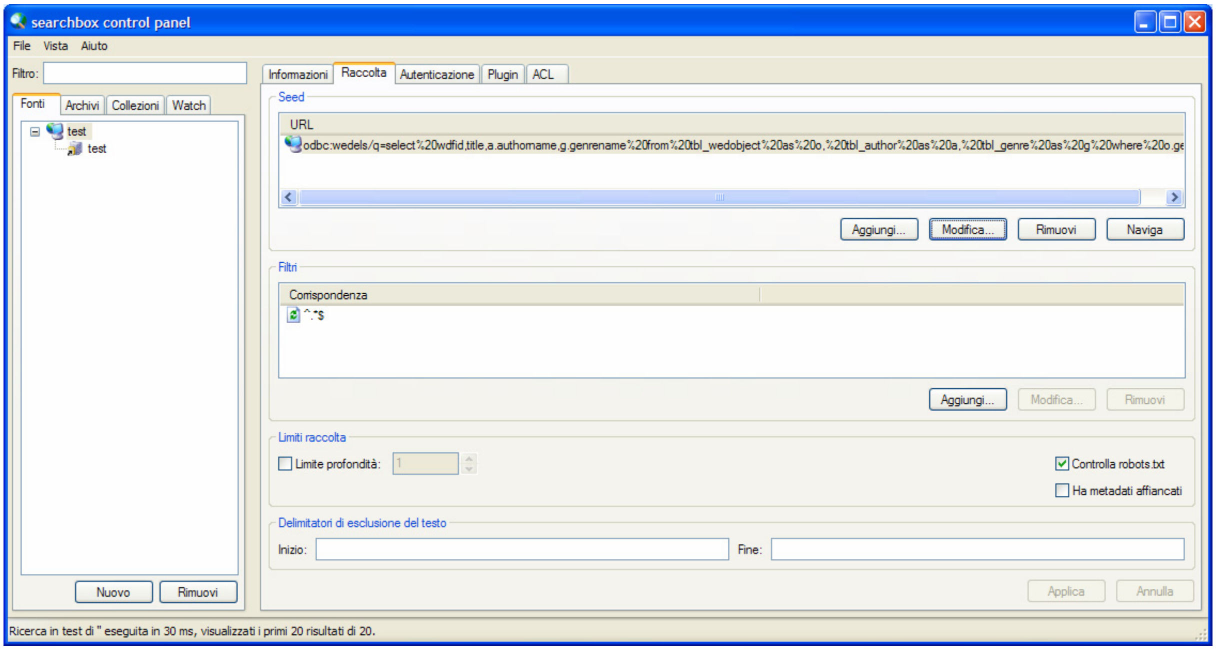
Fast ESP (www.fast.no/us/esp/)
It is the main competitor of Google. Fast has been recently acquired by Overture, which has been acquired by Yahoo! together with Altavista. Fast technology now replaces Google in the Yahoo! portal.

Some other good Open Source projects can be found on the web but none of them has already reached the state of a mature product. For a good reference, see the DMOZ directory at the address
http://dmoz.org/Computers/Software/Internet/Servers/Search/

4.3 Work Done: Integration and Modification/Extensions

Since the crawling and indexing are realized with an already existing tool (focuseek searchbox) no development has been done. However some testing has been done on the tool to import data from odbc database and mysql databases.

The figure shows the configuration of a source to gather information from a mysql database using an odbc interface:


searchbox control panel
File Vista Aiuto
Filtro:
Fonti Archivi Collezioni Watch
test
test
Nuovo Rimuovi
Informazioni Raccolta Autenticazione Plugin ACL
Seed
Aggiungi... Modifica... Rimuovi Naviga
Filtri
Aggiungi... Modifica... Rimuovi
Limiti raccolta
Delimitatori di esclusione del testo
Applica Annulla

URL
odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_wedobject%20as%20o,%20tbl_author%20as%20a,%20tbl_genre%20as%20g%20where%20o.ge

Corrispondenza
^.*$

□ Limite profondità: 1

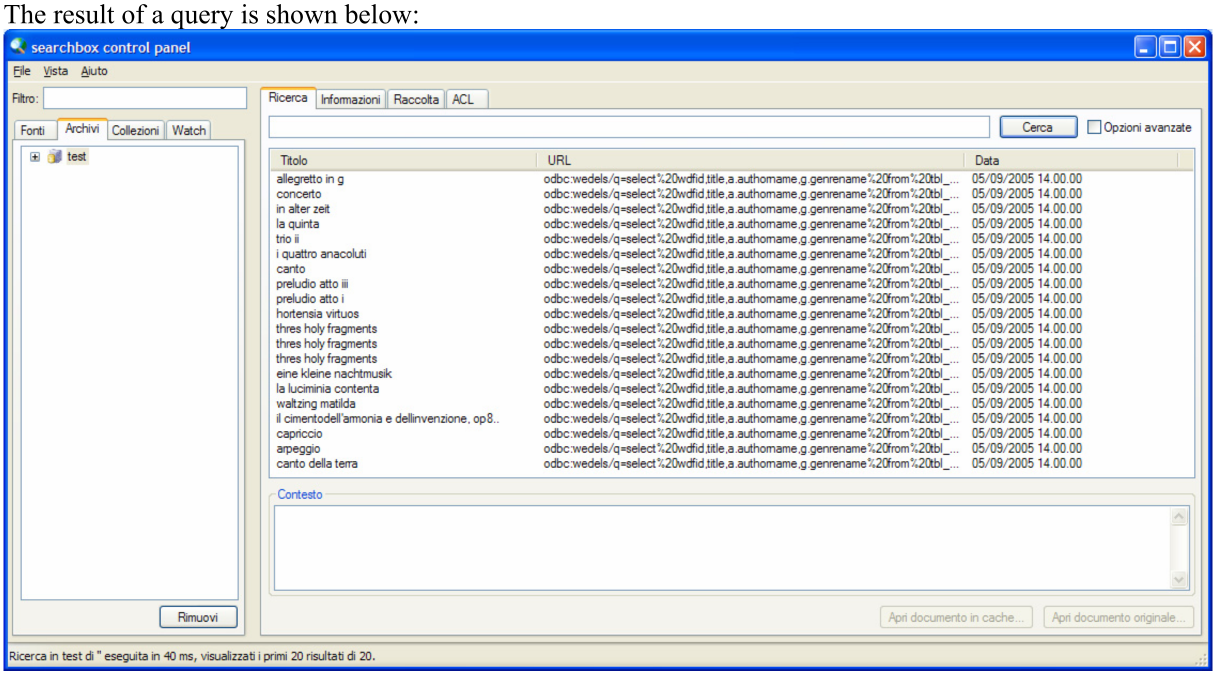
☑ Controlla robots.txt

□ Ha metadati affiancati

Inizio: Fine:

Ricerca in test di '' eseguita in 30 ms, visualizzati i primi 20 risultati di 20.

The result of a query is shown below:


searchbox control panel
File Vista Aiuto
Filtro:
Fonti Archivi Collezioni Watch
test
Rimuovi
Ricerca Informazioni Raccolta ACL
Cerca
Contesto
Apri documento in cache... Apri documento originale...

□ Opzioni avanzate

Titolo	URL	Data
allegretto in g	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
concerto	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
in alter zeit	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
la quinta	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
trio ii	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
i quattro anacoluti	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
canto	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
preludio atto iii	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
preludio atto i	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
hortensia virtuos	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
thres holy fragments	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
thres holy fragments	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
thres holy fragments	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
eine kleine nachtmusik	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
la luciminia contenta	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
waltzing matilda	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
il cimentodell'armonia e dellinvenzione, op8..	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
capriccio	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
arpeggio	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00
canto della terra	odbc:wedels/q=select%20wdfid,title,a.authorname,g.genrename%20from%20tbl_...	05/09/2005 14.00.00

Ricerca in test di '' eseguita in 40 ms, visualizzati i primi 20 risultati di 20.

5 Estimating Fingerprint for Indexing - Content Descriptors

From a human point of content descriptors are required for efficient and effective content retrieval of content databases. Traditionally this annotation was created by human experts. The human experts analyse the content and extract information from its interpretation. This information is used to classify the content according to a domain specific taxonomy. However, this traditional way of creating this information is time consuming and therefore costly.

Different methods for automatic creation of these content descriptors have been developed. Depending on the resulting content descriptors two different categories can be distinguished:

- Low-level descriptors are closer related to the information signal than to the content itself. In other words, the content is decomposed in different signals, which are analysed. Thus, low-level descriptors provide little or no meaning to end-users: The content descriptors describe content properties that are not directly perceived by humans.

  Different level of extraction granularity can be distinguished:
  1. single sample/signal value
  2. small regions/short signal window
  3. bigger regions/longer signal window

- High-level descriptors are content descriptors that can be perceived by humans. They can be calculated from the content or from the low-level descriptors.

  Different
  1. Syntactic descriptors can be understood without signal processing knowledge. Yet, they do not provide a semantic description of the content.
  2. Semantic descriptors are meaningful for humans but can also vary in their concreteness.

Obviously, semantic high-level features are the first choice for content descriptors that are used by humans. Unfortunately there is no clear separation between high- and low-level features. Additional high-level descriptors are not available for all kinds of formats.

6 Audio Content Descriptors

The creation of huge databases of audio content coming from both restoration of existing analog archives and new content is demanding fast and more and more reliable tools for content analysis and description, to be used for searches, content queries and interactive access. In that context, MPEG-7 has normalized a number of low-level descriptors to help characterizing audio titles for retrieving. Yet such descriptors make no sense for human users and cannot be used for intuitive navigation. Higher-level descriptors such as musical genres are of a greater interest since they have been widely used for years to organize music catalogues, libraries and record shops.

The automatic music genre recognition problem is of a particular interest in the context of audio context description since it shares a number of similarities with other applications in which one aims at extracting automatically *semantic* descriptors of audio content (such descriptors may for example characterize instrumentation or *moods* and may be associated to keywords such as *dark, simple, brassy, soft…*). As a matter of fact, based on the architecture of a genre recognizer, one can derives a number of other extractor of semantically meaningful descriptors. We thus focus in the following on the genre recognition problem as a typical pattern recognition problem of the music information retrieval community. Moreover, to characterize a high-level descriptor such as genre, one needs to extract a number of lower-level descriptors so that the following description focused on genre will review the complete state-of-the-art in audio content description.

6.1 State of the Art

Musical genres are the main top-level descriptors used by music dealers and librarians to organize their music collections. Though they may represent a simplification of one artist's musical discourse, they are of a great interest as summaries of some shared characteristics in music pieces.

With Electronic Music Distribution (EMD), music catalogues tend to become huge (the biggest online services propose around 1 million tracks [1]); in that context, associating a genre to a musical piece is crucial to help users finding what they are looking for.

In fact, the amount of digital music data urges for efficient ways to browse, organize and dynamically update collections: it definitely requires new means for automatic annotation. In the case of music genre annotation, Weare [2] reports that the manually labeling of hundred thousand songs for Microsoft's MSN Music Search Engine needed about 30 musicologists for one year.

At the same time, even if terms such as *jazz*, *rock* or *pop* are widely used, they remain poorly defined concepts so that the problem of automatic genre classification becomes a non-trivial task. In the second section of this survey, we discuss the importance of musical genres with their definitions and hierarchies. The third section presents low-level techniques to extract meaningful information from audio data to characterize musical excerpts. We then review the state of the art in genre classification through three main paradigms: expert systems, unsupervised classification, and supervised classification.

6.1.1 MUSICAL GENRES

Musical genres are categories that have arisen through a complex interplay of culture, art and market forces to characterize similarities between musicians or compositions and organize music collections. Fabbri [3] attempts to define musical genres based on a combination of formal, technical, semiotic, behavioral, social, ideological, economical and juridical rules. The boundaries between genres as well as their definition still remain pretty fuzzy making the problem of automatic classification a non trivial task; yet, music catalogues are part of our life and culture.

The music genre classification problem asks for a taxonomy of genres i.e. a hierarchical set of categories to be mapped onto a music collection (though artists may not like the idea of being classified into a category). Pachet and Cazaly [4] studied a number of musical genre taxonomies used in industry and on the Internet and showed that it is not straightforward to build up such a hierarchy of genres. As a good classification relies on a carefully thought taxonomy, we summarize here a number of question and difficulties.

- Artists, albums or titles?

One question to be raised is to what kind of musical item genre classification is to apply: to a title, to an album or to an artist.

If we assume that one song may be classified into one genre (which is discussable), it is not that simple for an album. Consider for example the White Album of the Beatles that contain acoustic folk songs ("Rocky Raccoon"), brutal hard-rock songs ("Helter Skelter"), experimental compositions ("Revolution 9"), British blues songs ("Yer blues").

The same remark applies to artists. Some of them have covered such a wide range of genres that it does not make sense to try to associate them to a specific class. Take the example of Frank Zappa who has recorded contemporary pieces for orchestra ("The Yellow Shark"), doo-wop ("Cruising with Ruben & The Jets"), jazz-rock ("Hot rats") or strange objects that may be considered as experimental rock ("Uncle Meat").

- Non-agreements on taxonomies

Pachet and Cazaly [4] have showed that there is no general agreement on genre taxonomies. Taking the example of **http://www.allmusic.com** (531 genres), **http://www.amazon.com** (719 genres) and **http://www.mp3.com** (430 genres), they only found 70 words common to the 3 taxonomies. They notice that some widely used terms like *rock* or *pop* denote different sets of songs and those hierarchies of genres are differently structured from one taxonomy to the other.

- Ill-definition of genre labels

If we take a close look at some specific and widely used musical genres, we observe how varied can be the criteria that define a specific genre:

- indian music is a geographically defined genre;
- baroque music is related to a period of time (while encompassing a wide range of styles and a wide geographic region);
- barbershop music is defined by a set of precise technical requirements;
- post-rock is a term devised by music critic Simon Reynolds...

Pachet and Cazaly [4] argues that this semantic confusion within a single taxonomy may lead to redundancies that may not be confusing for human users but that may hardly handled by automatic systems. Furthermore, genre taxons may be dependant on cultural references. For example, a song by french singer Charles Aznavour would be considered as "Variety" in France but would be filed as "World Music" in the UK.

- Scalability of genre taxonomies

Hierarchies of genres should also consider the possibility to add new genres to take into account music evolution. New genres appear frequently (trip-hop, acid-jazz, post-rock) and are typically the result of some merging of different genres (psychobilly can be seen as the merging of rockabilly and punk) or the splitting of one genre into subgenres (original hip-hop has lead to different subgenres such as gangsta rap, turntablism, conscious rap...).

It is a major issue for automatic system. Adding new genres and subgenres to a taxonomy is possible but having an automatic system requiring supervised training to adapt itself is more tricky. In any case, an automatic system needs to be thought in a hierarchical way so that the process of expending a genre taxonomy both in width (new root genres) and depth (subgenres) is equivalent to add a new classification tree or new leaves to an existing tree.

- Local conclusion

Due to the difficulty of building a universal taxonomy one has to consider a more reasonable goal. Actually, Pachet and Cazaly eventually gave up their initial goal of defining a general taxonomy of musical genres [4] and Pachet and al. decided to use a simple two-level genre taxonomy of 20 genres and 250 subgenres in the context of the Cuidado music browser [5].

In the case of automatic genre classification, most authors have used only a very reduced set of genres (between 3 and 10; notice that the genres chosen are more dependent on the data available to the researchers than on carefully defined taxonomies).

As we're going to see, the state of the art in automatic classification does not allow yet very refined genres hierarchies (at the moment it is not easy to distinguish "rock" from "pop" so it will be even harder to distinguish subcategories such as "surf music" and "garage rock").

6.1.2 FEATURES EXTRACTION

In the digital world, audio information is represented by bits allowing a direct reconstruction of an analogue waveform. Music information may also be described more or less rigorously by some higher-level model-based representation – typically some event-like formats such as MIDI or symbolic formats such as MusicXML.

However, in real world applications a precise symbolic representation of a (new) song is rarely available and one has to deal directly with audio samples. Audio samples, though they encode accurately the audio waveform, can not be used directly by automatic analysis systems because of the amount of data they represent and because of the little information contained in audio samples taken independently. The first step of analysis systems is thus to extract some *features* from the audio data to manipulate more meaningful information and to reduce the further processing.

Extracting features is the first step of most pattern recognition system. Indeed, once significant features are extracted, any classification scheme may be used. In the case of audio signals, features may be related to the main dimensions of music including melody, harmony, rhythm, timbre and spatial location. As a matter of fact when dealing with music one should also consider the presence of singing voice and the content of the eventual lyrics (though keyword extraction from singing segments in polyphonic music is pretty far from the state-of-the-art).

We describe here the main features used in general applications of MIR with a focus on those that make sense when modelling genre.

- Timbre

Timbre is currently defined in the literature as the perceptual features that make two sounds having the same pitch and loudness sound different, Features characterizing timbre describe spectral distribution of the signal though some of them are computed in the time domain. These features are global in the sense that they integrate the information of all sources and instruments in the music,

Most of these descriptors are to be computed at regular time interval over short windows of signal of length typically between 10 and 30 ms. In the context of classification, timbre descriptors are often summarized by evaluating low-order statistics of the descriptors' distribution over larger windows (called texture windows in [6] or bags of frames in [7]). Not only does it reduce further computations but it is also perceptually more relevant to model timbre on a higher time scale as the short frames of signal used to evaluate features are not long enough for human listeners to perceive something (as a matter of fact, short frames are used so that one can consider the signal stationary so that it is more coherent to use standard signal processing tools),

The impact of the size of the texture window over classification accuracy has been studied in [6]. It is shown that the use of a window does indeed increase significantly the classification accuracy compared to the direct use of the analysis frames. They conclude that texture windows of 1 second are a good compromise since no significant gain in classification accuracy is obtained by taking larger windows while the accuracy decreases (almost linearly) as the window is shortened.

We enumerate here a number of descriptors commonly used to characterize timbre. Most of them come directly from the speech recognition literature [8] while some others have been normalized in the context of MPEG-7 [9].

- Zero-Crossing Rate [6], [10]: it is defined as the number of zero-crossings of the signal in the time domain. It is a measure of noisiness of the signal and is correlated with pitch.
- RMS [10]: the simple root mean square energy of the signal frame,
- Loudness [10]: simple models of loudness consist typically of an exponentiation of the energy of the frame.
- Low Energy Feature [6], [10]: it is the percentage of frames within a larger window that have a RMS energy lower than the mean RMS energy across the window.
- Linear Prediction Coefficients [11], [12]: linear prediction has been designed in the context of speech recognition to model sound production: the observed sound is supposed to be the result of the linear filtering of a simple signal. The estimated coefficients of the filter can be used as timbre descriptors since they encode the effect of the resonating body of the instrument (or of the vocal track in the case of speech production).
- FFT coefficients [12], [13], [14]: the feature vector is simply the vector of the FFT coefficients.
- Mel-Frequency Cepstrum Coefficients [6], [7], [10], [11], [15]: Mel-Frequency Cepstrum Coefficients (MFCCs) are perceptually motivated features obtained by taking the log-amplitude of the magnitude spectrum, warping the spectrum onto a perceptual frequency scale, the Mel frequency scale and by applying a discrete cosine transform on the Mel coefficients so as to decorrelate the resulting feature vector. 13 coefficients are tycically used for speech recognition. The number of MFCCs is a discussed parameter in the music analysis community [16].
- Spectral Centroid [6], [10]: describes the center of gravity of the power spectrum. It is an economical description of the shape of the power spectrum. It indicates whether an audio spectrum is dominated by low or high frequencies and, additionally, it is correlated with a major perceptual dimension of timbre; i.e.sharpness.
- Spectral Roll-Off [6], [10]: spectral roll-off is a measure of spectral shape. It is defined as the frequency below which most of the power spectrum is concentrated (typically 85 % of the power spectrum).
- Spectral Flux [6], [10]: spectral flux is a measure of the amount of local spectral change. It is evaluated as the difference between the normalized magnitudes of successive spectral distribution.
- Spectrum Spread [10]: spectrum spread describes the second moment of the power spectrum. It is an economical descriptor of the shape of the power spectrum that indicates whether it is concentrated in the vicinity of its centroid, or else spread out over the spectrum. It allows

differentiating between tone-like and noise-like sounds.

- Spectrum Flatness [10]: spectrum flatness expresses the deviation of the signal's power spectrum over frequency from a flat shape (corresponding to a noise-like or an impulse-like signal). A high deviation from a flat shape may indicate the presence of tonal components.
- Harmonic Ratio [10]: harmonic ratio is loosely defined as the proportion of harmonic components within the power spectrum. It is derived from the correlation between the signal and a lagged representation of the signal, lagged by the fundamental period of the signal.
- Wavelet [17]: the wavelet decomposition scheme allows a division of the signal into octave sub-bands while providing good time and frequency resolution. The Daubechies Wavelet Coefficients Histograms (DWCHs) proposed in [17] are histograms of wavelet coefficients over a large window from which features are extracted by evaluating moments.

- Melody, Harmony

Harmony may be loosely defined as the use and study of pitch simultaneity and chords, actual or implied, in music. On the contrary, melody is a succession of pitched events perceived as a single entity. Harmony is sometimes referred to as the vertical aspect of music with melody being the horizontal aspect,

Melodic and harmonic analysis having been for a long time used by musicologists to study musical structures, it is tempting to try to integrate such analysis when modeling genre. Though rather evolved analysis algorithms have been proposed when dealing with symbolic MIDI data where the pitch and time position information is known, the state-of-the-art of melodic and harmonic analysis of unrestricted audio signals is far less advanced because of the difficulty to extract reliably note onsets, pitch and chords.

A good overview of melody description and extraction in the context of audio content processing may be found in [18]. For the estimation of multiple fundamental frequencies of concurrent musical sounds, one may refer to [19] while chord extraction is addressed in [20]. Analogously to the case of speech recognition in which models of words and language are used to improve performances, use of higher-level knowledge such as *typical* chord progressions may greatly improve analysis results.'

In any case, for the time being, melodic and harmonic content are more robustly described by lower level attributes than notes or chords. To our knowledge, there has been only one attempt to use pitch features when modeling genres of audio signals [6].

The idea is to use a function characterizing pitch distribution of a short segment like most melody/harmony analyzer; the difference is that no decision on the fundamental frequency, chord, key or other high-level feature is undertaken. On the contrary, a set of descriptors are computed from this function including amplitude and positions of its main peaks, interval between peaks, sum of the detection function and possibly any kind of statistical descriptor of the distribution of the pitch content function. Two versions of the pitch function are used: an *unfolded* version that contains information about the pitch range of the piece and a *folded* one in which all pitches are mapped to a single octave giving a good description of the harmonic content of the piece.

- Rhythm

A precise definition or rhythm does not exist. Yet most authors refer to the idea of *temporal regularity*, As a matter of fact, the perceived regularity is distinctive of rhythm and distinguishes it from *non-rhythm*. More generically, the word rhythm may be used to refer to all of the temporal aspects of a musical work.

Intuitively, it is clear that rhythmic content may be a dimension of music to consider when trying to discriminate between *straight-ahead* rock music from rhythmically more complex latin music for example, or to isolate some classical music in which the sensation of pulse is not so evident and the expressive rhythm variation more common.

Here again rhythmic content analysis of symbolic MIDI data has preceded analysis of real world audio. A review of automatic rhythm description systems may be found in [21]. These automatic systems may be oriented towards different applications dedicated to rhythm: tempo induction, beat tracking, meter induction, quantization of performed rhythm, or characterization of intentional timing deviations.

In [22], Gouyon et al. study genre classification with a focus on the use of tempo. They show that by using manually assigned tempo values, genre classification may be improved but that with automatically induced tempi, a lower level approach leads to a better classification (see [23]): the state-of-the-art tempo tracking algorithms typically make errors of metrical levels (e.g. they often output multiples or sub-multiples of the correct tempo).

The same authors report on genre classification based on rhythmic patterns [24] obtained by averaging amplitude envelopes of the signal between measure bars. Though they show that rhythmic patterns help at improving classification results, the measure bars needed for their experiences are obtained semi-automatically by running a beat-tracking algorithm and selecting the measure bars by hand. Given the state-of-the-art accuracy of measure bars detection algorithm (see [25]), it is reasonable to think that here again a lower level approach would be best suited for the moment.

Following the same approach as the one introduced for low-level pitch attributes, descriptors may be extracted from a function measuring the importance of periodicities in the range of perceivable tempi (typically 40 to 200 bpm in genre classification applications). Such function may be obtained by autocorrelation-like transform of features over time [25] (interesting features being usually energies in different frequency bands). It is also possible to use FFT to evaluate modulations of features [26] or to build a histogram of inter-onset intervals [27].

Gouyon et al. [23] give an in-depth study on low-level rhythmic descriptors extracted from different periodicity representations. In particular, they obtain encouraging results with a set of MFCCs-like descriptors computed from a periodicity function (rather than from a spectrum).

- Extracting features from semantically significant audio segments

The descriptors presented earlier may be extracted for the complete audio signals. Yet, in many classification tasks, a small segment of audio is used as it may contain sufficient information to characterize the content of a complete song because of the repetitions in musical structure observed in many musical genres. This idea is all the more significant that the required computations may be greatly reduced by considering only a small part of the signal.

Most of the proposed algorithms for musical genre classification use indeed one small segment of audio per title: typically a 30 seconds long segment starting 30 seconds after the beginning of the piece to avoid introductions that may not be representative of the whole piece.

In the context of artist identification, Berenzweig et al. [28] have proposed to detect automatically singing segments and have obtained improved results by analyzing only the singing part: it may indeed be easier to identify artists by listening to their voices rather than their music.

Of course, this only applies to titles with singing voice but a more general approach based on recent research on automatic summarization of titles [29] may help genre characterization: for example, the most repeated part of a title (typically the chorus in popular music or the theme in jazz) could be identified and analyzed in term of genre rather than a random part of the signal which may be less representative. Yet, as far as known no such approach has been evaluated in the context of genre classification.

- Cultural features

Alternatively some authors have proposed to take some cultural features into account arguing that there may not be sufficient information in audio signals to characterize the musical genre of a title. Indeed, when one tries to derive genre from audio only, he assumes that genre is an intrinsic attribute of a title as its tempo for example, which is definitely arguable (see section II).

The proposed approach is thus to mine the web to find keywords associated to the artist names. Pachet et al. [30] use this technique in the context of unsupervised clustering of a music collection (i.e. without explicitly associating a genre label to a title) while in [31], keywords are used to characterize musical genres in a supervised fashion. In [32], cultural features are used to improve standard genre characterization based on audio features: in this case cultural features help at classifying properly acoustically dissimilar music within the same genre and similar music belonging to different genres.

- Local conclusion

Extraction of high-level descriptors from unrestricted polyphonic audio signals is not yet state of the art. Thus most approaches focus on timbre modeling based on combinations of low-level descriptors. Timbre may contain sufficient information to characterize roughly musical genres as research has found that humans with little to moderate musical training were able to perform a correct classification of music (among 10 genres) in 53 % of the cases after listening to only 250 milliseconds and in 72% of cases based on only 3 seconds of audio [33]. This suggests that no high-level understanding of music is needed to characterize genres as 250 milliseconds and in a lesser manner 3 seconds are too little time to recognize a musical structure.

Aucouturier and Pachet [34] have a more pessimistic point of view. They have studied the correlation between timbre similarity and genre. They used a state-of-the-art timbre similarity measure [35] and a database of 20000 titles distributed over 18 genres. Their results show that there is only little correlation between timbre and genres suggesting that classification schemes based solely on timbre are intrinsically limited. They also suggest that such classification schemes may hardly scale in both the number of titles and in the number of genre classes.

6.1.3 EXPERT SYSTEMS

Expert systems explicitly implement sets of rules. For the genre classification task, this would be equivalent to enumerate a number of rules that would characterize precisely and uniquely a genre. As far as we know, no model based on expert systems has been proposed to characterize musical genres. The work by Pachet and Cazaly [4] on a taxonomy of musical genres can be compared to an expert system approach though it did not lead to an actual implementation. Yet it is worth mentioning as it allows a deeper comprehension of the difficulties of music genres classification.

Pachet and Cazaly have tried to define characteristics of genres and their relations. They have formally stated differences among genres with a language based on descriptors such as the instrumentation, the type of voice, the type of rhythm, the tempo of the song (for example, *Ska* is derived from *Mento* and is different because it has a faster tempo and has a brass section). This implies that these descriptors must be detailed enough to characterize subgenres differences.

This approach, if possible at all, is not appropriate for genre classification given the complexity of the task and the difficulty to objectively describe very specific subgenres. Moreover it requires a (automatic) manner to obtain reliable high-level descriptors from the audio signal, which is not state-of-the-art as seen in the previous section.

Expert systems, though they incorporate deep knowledge of their subject, are expensive to build and to maintain. As the number of manually generated rules grows, they may yield unexpected interactions and side effects, so that software engineering issues become increasingly important, In the last few years, the machine learning approach has garnered increasing interest. From the point of view of related disciplines, the machine learning approach has come to dominate similar areas of natural language processing and pattern recognition such as automatic speech recognition or face recognition. Finally from the point of view of objectively evaluated performance, systems based at least partly on machine learning have outperformed those based purely on expert systems in recent quantitative evaluations.

6.1.4 UNSUPERVISED CLASSIFICATION

While other approaches tend to classify music given an arbitrary taxonomy of genres, another point of view is to cluster the data in a non-supervised way so that a classification will emerge from the data based on objective similarity measures. The interest is to avoid the constraint of a fixed taxonomy, which may suffer from ambiguities and inconsistencies as it has been seen earlier, Moreover, some titles may simply not fit into a given taxonomy.

In the unsupervised approach, an audio title is represented by a set of features as seen in section III and a similarity measure is used to compare titles one with another. Unsupervised clustering algorithms then take advantage of the similarity measure to organize the music collection with clusters of similar titles.

- Similarity measures

The simplest choice to measure distance between two feature vectors is to use a Euclidean distance. It is used in [26] in the context of unsupervised clustering of a music collection by sound similarity. Another popular distance measure is the cosine distance used in [36].

Notice however that these distances will only make sense if the feature vectors are time-invariant. If not two perceptually similar titles may be distant according to the measure if the similar features are time-shifted. One solution to build a time-invariant representation of a time series of feature vectors is to first build a statistical model of the distribution of the features and use the distance to compare models directly. Typical models include K-means, Gaussian and Gaussian mixtures (GMMs).

The Kullback-Lleiber divergence or relative entropy is the natural way to evaluate distance between probability distributions but is not suited for GMMs. Alternative measures include sampling [35], Earth's Mover distance [37] and the Asymptotic Likelihood Approximation [38].

Considering the fact that, unlike most classic pattern recognition problems, the data to be classified are time series data, Shao et al. [11] use Hidden Markov Models (HMMs) to model the relationship between features over time. One interest of HMMs is that they provide a proper distance metric [8] so that once each piece is characterized by its own HMM, the distance between any pieces of the database can be computed.

- Clustering algorithms

K-means is probably the simplest and most popular clustering algorithm [39]. It allows for partitioning a set of vectors into K disjoint subsets. One of its weaknesses is that it requires the number of clusters (K) to be known in advance,

Shao et al. [11] cluster their music collection with the Agglomerative Hierarchical Clustering [40], a clustering algorithm that starts with *N* singleton clusters (where *N* is the number of titles of the database) and that forms a sequence of clusters by successive merging. When the number of desired clusters is known, the merging process will be stopped when this number of cluster is reached while if it is unknown, the merging process will be stopped when the distance between two nearest clusters is above a threshold.

The Self-Organizing Map (SOM) [41] and the Growing Hierarchical Self-Organizing Map (GHSOM) [42] are used to cluster data and organize it on 2-dimensional space in such a way that similar feature vectors are grouped close together. SOMs are unsupervised artificial neural networks that map high dimensional input data onto lower dimensionality output spaces while preserving the topological relationships between the input data items as faithfully as possible. GHSOMs are a special case of SOMs which make use of a hierarchical structure of multiple layers where each layer consists of a number of independent SOMs. Rauber et al. [26] use an output space of dimension 2 to allow for a visual representation of a music collection with a GHSOM.

- Local conclusion

In some terms, the major drawback of unsupervised techniques can be that the obtained clusters are not labeled. In any case, the obtained clusters do not always reflect genre hierarchies, rather similarities dependent on the type of features (rhythmical similarities, melodic similarities, etc.).

Rousseaux and Bonardi [43] argue anyway that in the context of EMD, the notion of genre may disappear as develops an *ad-hoc* organization of audio sample centred on prototypes and similarity. Up to a certain extent, we agree that some forward thinking leads to a more "perceptual" approach to music classification, based at the same time on the combination of multiple labels and *cross-genre* references: people like different styles of music, and what they like in each one may share well identified features.

SUPERVISED CLASSIFICATION

The supervised approach to music genre classification has been studied more extensively. These methods suppose that a taxonomy of genres is given and they try to map a database of songs into it by machine learning algorithms. As a first step, the system is trained with some manually labeled data, and then it is used to classify unlabelled data.

We describe here a number of commonly used supervised machine learning algorithms. We do not pretend to make an exhaustive list of such algorithms but focus on those that have been used in the context of music genre classification. We then present the results obtained with these algorithms in the literature.

- Supervised classifiers
    1. K-Nearest Neighbor (KNN): it is a non-parametric classifier based on the basic idea that a small number of neighbours influence the decision on a point. More precisely, for a given feature vector in the target set, the K closest vectors in the training set are selected (according to some distance measures) and the target feature vector is assigned the label of the most represented class in the K neighbours. Of course the computing time goes up as K goes up but the advantage is that higher values of K provide smoothing that reduces vulnerability to noise in the training data. It has been proven that the error of KNN is asymptotically at most twice as large as the Bayesian error rate. KNNs are evaluated in the context of genre classification in [44], [17], [6].
    2. Gaussian Mixture Models (GMM): GMMs model the distribution of feature vectors. For each class, we assume the existence of a probability density function expressible as a mixture of a number of multi-dimensional Gaussian distributions. The iterative Expectation Maximization (EM) algorithm is used to estimate the parameters for each Gaussian component and the mixture weights. GMMs have been widely used in the music information retrieval community, notably to build timbre models as seen in section (V. A.). They have been used to model directly musical genres in [17], [6], [7],
    3. Linear Discriminant Analysis (LDA): LDA is known to learn discriminative feature transformations very well. The basic idea of LDA is to find a linear transformation that best discriminates among classes and perform classification the transformed space based on some metric such as Euclidean distances. LDA has been tested for musical genre classification in [17], [45], [7].
    4. Support Vector Machines (SVM): SVMs have shown excellent results for data classification and regression [46]. Their success in practice is based on two properties: margin maximization (which allows for a good generalization of the classifier) and nonlinear transformation of the feature space with kernels (as a data set is more easily separable in a high dimensional feature space), SVMs have been notably used in the context of genre classification by [17], [45].
    5. Maximal Binary Classification Tree: West and Cox [7] use a Maximal Classification Binary Tree built by forming a root node containing all the training data and then splitting that data into two child nodes by using either a LDA or a single Gaussian classifier with Mahalanobis distance measurements. To split a node, all possible combinations of classes are formed and the combination of classes yielding the best split is chosen. The maximal tree, over-fitting the training set is evaluated before a pruning the tree back to a more sensible size. Notice that the building of the tree is unsupervised while the classifiers used for splitting on each node are trained in a supervised manner.
    6. Artificial Neural Network (ANN): An ANN is an information-processing paradigm inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. The most widely used supervised ANN for pattern recognition is the Multi-Layer Perceptron (MLP). It is a very general model that can in principle approximate any non-linear function. A combination of three different MLPs is used in [44] to characterize musical genres. The idea is use different segments of the audio stream and to combine the classification results to increase overall accuracy. A MLP is used in [28] for a close problem, artist identification; the architecture used takes advantage of contextual information by inputting to the network 5 adjacent feature vectors (this architecture is also referred to in the literature as the Feedforward Time-Delay Neural Network, TDNN). A TDNN is also used in [32].
    7. Explicit Time Modelling with Neural Networks (ETM-NN): Neural Networks, as well as the other reviewed architectures, can only handle static patterns, Some architectures oriented towards the processing of temporal sequences have been proposed (Recurrent Networks such as the Elman-Network [47] but have not been used yet in the context of music genre classification. Soltau et al. [15] have introduced in the context of recognition of music genres an original method for explicit time modeling of temporal structure of music (ETM-NN). In this architecture, a multi-layer perceptron is trained to recognize given input feature vectors that are

0.4 seconds long. The main idea is to use the hidden layer of the perceptron and not its output, which is supposed to give the genre. As a matter of fact, it is known that the first half of a feed-forward network performs a specific nonlinear transformation of the input data into a space in which the discrimination should be simpler. The activation of these hidden neurons corresponds to the use of a compact representation of the input feature vector. Each hidden neuron can be understood as an abstract musical event – not necessarily related to an actual musical meaning. An abstract event e_i occurs if the hidden unit *i* has the highest activation of all hidden units. The sequence of abstract events over time is then analysed to build one single feature. More specifically, the number of events e_i (unigram), the number of pairs e_i e_j (bigram) and the number of triplets e_i e_j e_k (trigram) are evaluated as well as the event durations, the mean, the maximum and the variance of event activations. All of these features, normalized over the length of the sequence are combined into a single vector that is given to another neural network. This second network implements the final decision about the genre of the musical piece.

8. Hidden Markov Model (HMM): Hidden Markov Models (HMMs) already introduced in section earlier as a similarity measure can also be used for classification purposes. They have been extensively used in speech recognition [48] because of their capacity to handle time series data. HMMs may be understood as a doubly embedded stochastic process: one process is not observable (hidden) and can only be observed through another stochastic process (observable) that produces the time set of observations. A HMM is defined by its number of states, the transition probability between its states, the initial state distribution and the observation probability distribution. Mixtures of Gaussians typically model observation probability densities. This implies a strong assumption on the distribution of the emission probability that may be relaxed by modelling emission probabilities with neural networks (see [49]). HMMs may have a low discriminative power because they are usually trained with a maximum likelihood criterion rather than with an optimal maximum a-posteriori criterion. In other words, during the training, the likelihood that the model of a genre did produce an observation is maximised but the likelihood of the other models is not minimized. A number of alternative solutions for the training of HMMs have been proposed to ensure them a better discrimination power (see [50]). Though they may be well suited to modelling music, to our knowledge, HMMs have only been used in [18] for genre classification of audio content (they have been used in [16] as well but in the case of unsupervised organization of a music collection).

Mixture of Experts (ME): A Mixture of Experts solves a classification problem by using a number of classifiers to decompose it into a series of sub-problems. Not only does it reduce the complexity of each single task but it also improves the global accuracy by combining the results of the different classifiers (experts). Of course, the number of needed classifiers is increased but having each of them a simpler problem to handle, the overall required computational power is reduced. Using a mixture of classifiers, each subtask may focus either on a subset of the attributes (feature selection), on different sample data (resampling, i.e. sub-sampling, bagging, boosting…), or on a different data labelling (decomposition of polychotomies into dichotomies). A range of solutions has been proposed in literature for the combination of different models into a global system. A possible solution is to use a majority vote of the different experts. This solution is used in [44] where 3 different MLPs characterize three different segments of a single song. Another possibility is to average the output probability estimates of each expert for each class. A more elaborated strategy is to consider the combination of the outputs of each expert as another learning problem. When each expert is working on a different feature set, the combination of results can be itself a function of the features so that inputs control the weights associated to each expert: such a method implements a sort of dynamic feature selection strategy. This latter classification scheme is used in our prototype for AXMEDIS with SVM expert and three different feature sets (timbre, energy and rhythm).

- Classification results

The taxonomy used in state-of-the-art works on genre classification is often very simple and incomplete (between 2 and 10 genres and rarely more than 1000 songs) and usually reflect more the data available to the authors than a rigorous analysis of genres; it is consequently rather difficult to compare the different approaches. The Music Information Retrieval Evaluation eXchange 2005 (MIREX) [51] is trying to unify efforts and give a rigorous comparison of algorithms.

- Local conclusion

The major interest of supervised classification compared to the expert system approach is that one does not need to describe explicitly musical genres: the classifier attempts to form automatically relationships between the features of the training set and the related categories.

An important point to consider in supervised approaches is how to expand the taxonomy of genres they are built upon. New genres appear frequently (*trip-hop*, *acid-jazz*, *post-rock*) and are typically the result of some merging of different genres (*psychobilly* can be seen as the merging of *rockabilly* and *punk*) or the splitting of one genre into subgenres (original *hip-hop* has lead to different subgenres such as *gangsta rap*, *turntablism*, *conscious rap*…). Hierarchical systems have to be favoured as the process of expending their genre taxonomy both in width (new root genres) and depth (new subgenres) is equivalent to adding a new classification tree or new leaves to an existing tree.

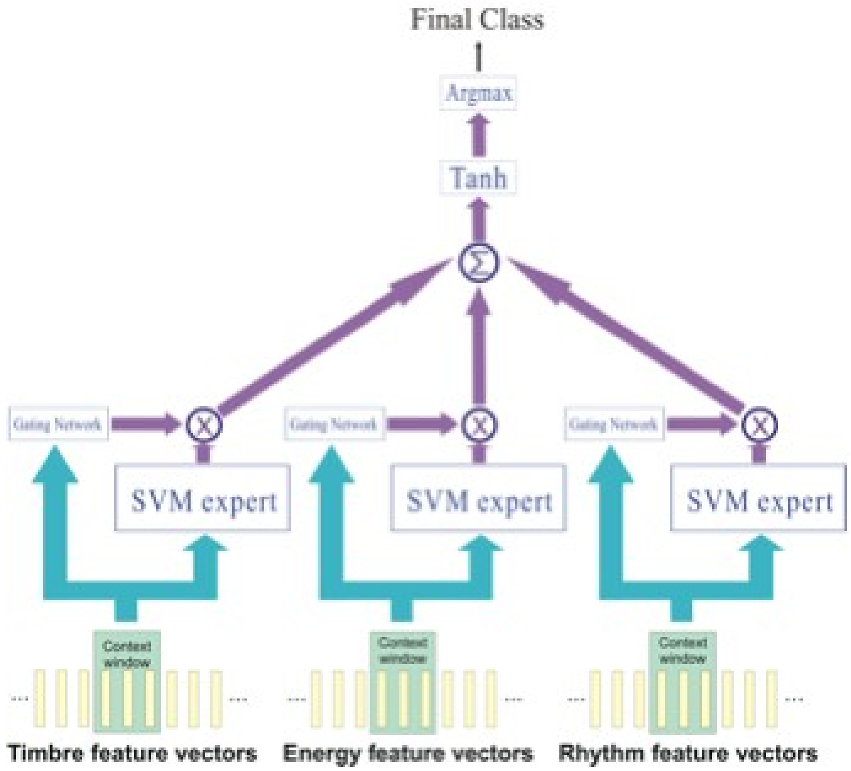
6.2 Problems

Three main difficulties emerge from works on audio content description:

1. Training data: In the context of machine learning, the more abundant the training data, the better. Yet freely available audio content is rare and does not represent accurately the variety of music composed up to now. One solution proposed in the context of music information research is to exchange standardised low-level descriptors of audio content between laboratories so that researchers can take advantage of the audio content of their colleagues (researchers usually using their personal CD collections).
2. Ground truth: In a classification task, the labels associated to an item are crucial. In the context of semantic description of audio content, labels may be difficult to obtain: as we have seen earlier, musical genres are ill-defined and ambiguous and it is all the more true for descriptors such as *moods*. In the context of AXMEDIS, we will use labels mined from the AllMusicGuide (**http://www.allmusic.com**) as a reference (genre, moods and instrumentation) and labels from the Epitonic collection for the data available from Epitonic (http://www.epitonic.com).
3. Evaluation of similarity measures: No rigorous evaluation of audio similarity measures has been proposed yet because of the lack of available data describing similarities between titles. In the context of MIREX [51], we will set up a database of similarity relations between artists based on the AllMusicGuide (**http://www.allmusic.com**) and Epitonic (http://www.epitonic.com).

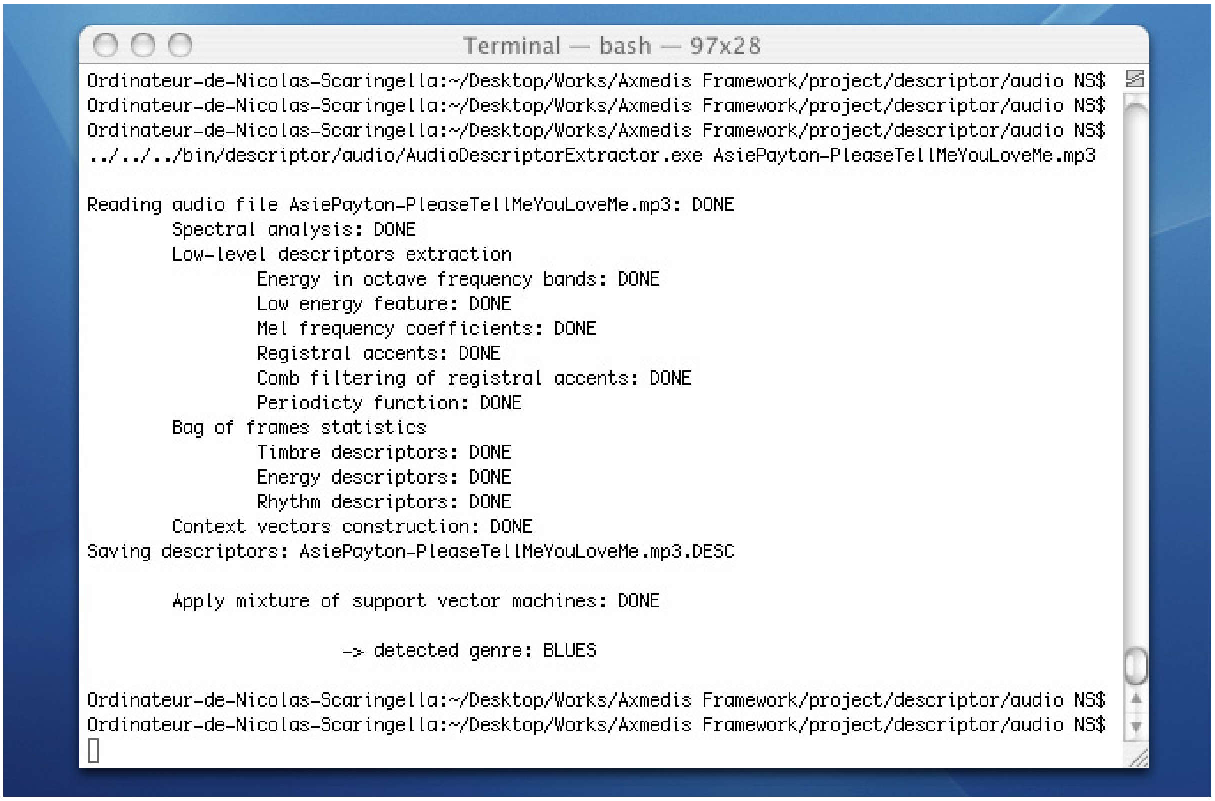
6.3 Work Performed

1. Database: Two databases have been set up to build two different genre models and evaluate them. The first one was built from personal CDs with genre labels mined from the AllMusicGuide (**http://www.allmusic.com**). The second database is made of the content freely downloadable on the Epitonic website (http://www.epitonic.com). In that case, genre labels were inferred from the labels used in the Epitonic catalogue. Since the content of Epitonic is freely downloadable (though not distributable), we have proposed to use it in the context of the MIREX [51] evaluation contest and is now widely used by other researchers in the field.
2. An algorithm has been implemented in C++ for the AXMEDIS audio content descriptors extraction: parameterizes audio content by extracting 3 sets of features describing 3 different dimensions of music: timbre, energy and rhythm. Once features are extracted, a mixture of SVMs is used for classification into musical genres. The following figure illustrates the architecture of the corresponding classification tool.



3. A prototype demonstrator running on MacOsX, MinGW32 (and possibly Linux though it has not been tested yet) has been posted on the CVS. The prototype allows training a classifier given a list of audio files and the labels to be associated with (music genre or other semantic descriptors). The prototype can be used to evaluate a trained model given another list of audio files and labels. Apart from these classifier training and testing mode, the prototype can be used to extract the high-level descriptors of a single file given a trained model. Audio content may be presented as compressed or uncompressed files in various formats including mp3, wma, wav… The following figure illustrates the command line interface of the tool: it reads an audio file, extract some descriptors, save them and evaluate the musical genre of the file.
4. The quality of the system for musical genre classification has been evaluated in the context of MIREX 2005 [51] and has been compared successfully to other state-of-the-art algorithms. Two sets of data were used (refer to [51]):
    a. the MAGNATUNE dataset composed of 1005 training files and 510 testing files distributed over 10 genres (ambient, blues, classical, electronic, ethnic, folk, jazz, new-age, punk, rock)
    b. the USPOP dataset composed of 940 training files and 474 testing files distributed over 6 genres (country, electronica & dance, new-age, rap & hip-hop, reggae. rock)

    The normalized overall classification accuracy is of 72.30 % for the MAGNATUNE dataset and of 77.67 % for the USPOP dataset. Detailed results may be found in [51].



```
Ordinateur-de-Nicolas-Scaringella:~/Desktop/Works/Axmedis Framework/project/descriptor/audio NS$
Ordinateur-de-Nicolas-Scaringella:~/Desktop/Works/Axmedis Framework/project/descriptor/audio NS$
Ordinateur-de-Nicolas-Scaringella:~/Desktop/Works/Axmedis Framework/project/descriptor/audio NS$
../../../bin/descriptor/audio/AudioDescriptorExtractor.exe AsiePayton-PleaseTellMeYouLoveMe.mp3

Reading audio file AsiePayton-PleaseTellMeYouLoveMe.mp3: DONE
        Spectral analysis: DONE
        Low-level descriptors extraction
                Energy in octave frequency bands: DONE
                Low energy feature: DONE
                Mel frequency coefficients: DONE
                Registral accents: DONE
                Comb filtering of registral accents: DONE
                Periodicty function: DONE
        Bag of frames statistics
                Timbre descriptors: DONE
                Energy descriptors: DONE
                Rhythm descriptors: DONE
        Context vectors construction: DONE
Saving descriptors: AsiePayton-PleaseTellMeYouLoveMe.mp3.DESC

        Apply mixture of support vector machines: DONE

                        -> detected genre: BLUES

Ordinateur-de-Nicolas-Scaringella:~/Desktop/Works/Axmedis Framework/project/descriptor/audio NS$
Ordinateur-de-Nicolas-Scaringella:~/Desktop/Works/Axmedis Framework/project/descriptor/audio NS$
```

6.4 References

[1] International Federation of the Phonographic Industry, "Digital music report 2005", published by *IFPI*, London, England, January 2005. Available: http://www.ifpi.org/site-content/library/digital-music-report-2005.pdf

[2] R. Dannenberg, J. Foote, G. Tzanetakis, C. Weare, "*Panel: new directions in music information retrieval*", in *Proc. Int. Computer Music Conf.*, Habana, Cuba, September 2001.

[3] F. Fabbri, "*A theory of musical genres: two applications*", in *Popular Music Perspectives*, ed. D. Horn and P. Tagg, 1981, Goteborg and Exeter: International Association for the Study of Popular Music, pp. 52-81.

[4] F.Pachet, D. Cazaly, "*A taxonomy of musical genres*", in *Proc. Content-Based Multimedia Information Access (RIAO)*, Paris, France, 2000.

[5] F. Pachet, J.J. Aucouturier, A. La Burthe, A. Zils, A. Beurive, "The Cuidado Music Browser: an end-to-end Electronic Music Distribution System", *Multimedia Tools and Applications*, 2004. Special Issue on the *CBMI03* Conference, Rennes, France, 2003.

[6] G. Tzanetakis, P. Cook, "*Musical genre classification of audio signals*", in *IEEE Transactions on Speech and Audio Processing*, Vol. 10, No. 5, July 2002.

[7] K. West, S. Cox, "*Features and classifier for the automatic classification of musical audio signals*", in *Proc. Of the 5th Int. Conf. on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[8] L. Rabiner, B.H. Juang, "*Fundamentals of speech recognition*", Englewood Cliffs, NJ, Prentice-Hall, 1993.

[9] ISO/IEC JTC 1/SC 29, "*Information Technology – Multimedia Content Description Interface – Part 4: Audio*".

[10] J.J. Burred, A. Lerch, "*A hierarchical approach to automatic musical genre classification*", in *Proc. Of the 6th Int. Conf. on Digital Audio Effects (DAFx)*, London, UK, 2003.

[11] X. Shao, C. Xu, M. Kankanhalli, "*Unsupervised classification of musical genre using hidden Markov model*", in *IEEE Int. Conf. of Multimedia Explore (ICME)*, Taibei, Taiwan, China, 2004.

[12] G. Tzanetakis, P. Cook, "*Audio Information Retrieval (AIR) Tools*", in *Proceedings of the 2000 International Symposium on Music Information Retrieval*, Plymouth, Massachusetts, USA, 2000.

[13] M. Talapur, S. Nath, H. Yan, "*Classification of musical genre*", Working Paper, Computer Science Department, Carnegie Mellon University, 2000.

[14] H. Deshpande, U. Nam, R. Singh, "*Classification of music signals in the visual domain*", in Proceedings of the 2001 Digital Audio Effects Workshop, 2001.

[15] H. Soltau, T. Schultz, M. Westphal, A. Waibel, "*Recognition of music types*", in *Proc. IEEE Int .Conf. on Acoustics, Speech and Signal Processing (ICASSP), Seattle, USA*, 1998.

[16] J.J. Aucouturier, F. Pachet, "*Improving timbre similarity: how high's the sky ?*", in *Journal of Negative Results in Speech and Audio Sciences*, 2004,

[17] T. Li, M. Ogihara, Q. Li, "*A comparative study on content-based music genre classification*", in *Proc. Of the 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Toronto, Canada, 2003.

[18] E. Gomez, A. Klapuri, B. Meudic, "*Melody description and extraction in the context of music content processing*", in *Journal of New Music Research*, Vol. 32(1), 2003.

[19] A. Klapuri, "*Multiple fundamental frequency estimation based on harmonicity and spectral smoothness*", in *IEEE Trans. Speech and Audio Proc.*, 11(6), 804-816, 2003.

[20] G. Zoia, R. Zhou, D. Mlynek, "*A multi-timbre chord/harmony analyzer based on signal processing and neural networks*", in *Proc. IEEE Int. Workshop on Multimedia Signal Processing*, Siena, Italy, 2004.

[21] F. Gouyon, S. Dixon, "*A review of automatic rhythm description system*", in *Computer Music Journal*, vol. 29, pp. 34-54, 2005.

[22] F. Gouyon, S. Dixon, "*Dance music classification: a tempo-based approach*", in *Proc. 5th Int. Conf. on Music Information Retrieval (ISMIR),* Barcelona, Spain*,* 2004.

[23] F. Gouyon, S. Dixon, E. Pampalk, G. Widmer, *"Evaluating rhythmic descriptors for musical genre classification*", in *AES 25th International Conference*, London, England, 2004.

[24] S. Dixon, F. Gouyon, G. Widmer, "*Towards characterization of music via rhythmic patterns*", in *Proc. Of the 5th Int. Conf. on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[25] A. Klapuri, A. Eronen, J. Astola, "*Analysis of the meter of acoustic musical signals*", in *IEEE Trans. Speech and Audio Proc.*, 2004.

[26] A. Rauber, E. Pampalk, D. Merkl, "*Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarity*", in *Proceedings of the 3rd International Conference on Music Information Retrieval*, Paris, France, 2002.

[27] S. Dixon, "*Automatic Extraction of Tempo and Beat from Expressive Performances*", in *Journal of New Music Research*, 30(1), pp 39-58, 2001.

[28] A. Berenzweig, D. Ellis, S. Lawrence, "*Using voice segments to improve artist classification of music*", in *Proc. of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, 2002.

[29] G. Peeters, A. La Burthe, X. Rodet, "*Toward automatic music audio summary generation from signal analysis*", in *Proc. 3rd Int. Conf. on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[30] F. Pachet, G. Westermann, D. Laigre, "*Musical data mining for electronic music distribution*", in *Proceedings of the 1st WedelMusic Conference*, Firenze, Italy, 2001.

[31] P. Knees, E. Pampalk, G. Widmer, "*Artist classification with web-based data*", in *Proc. Of the 5th Int. Conf. on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[32] B, Whitman, P. Smaragdis, "*Combining musical and cultural features for intelligent style detection*", in *Proc. 3rd Int. Conf. on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[33] D. Perrott, R. O. Gjerdingen, "*Scanning the dial: an exploration of factors in the identification of musical style*", Research Notes, Department of music, Northwestern University , Illinois, 1999.

[34] J.J. Aucouturier, F. Pachet, "*Representing musical genre: a state of the art*", in *Journal of New Music Research*, vol. 32, no. 1, pp. 83-93, 2003.

[35] J.J. Aucouturier, F. Pachet, "*Music similarity measures: what's the use?*", in *Proc. of the 3rd Intternational Symposium on Music Information Retrieval*, 2002.

[36] M. Cooper, J. Foote, "*Summarizing popular music via structural similarity analysis*", in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[37] B. Logan, A. Salomon, "*A content-based music similarity function*", in *Cambridge Research Laboratory Technical Report Series*, June 2001.

[38] N. Vasconcelos, "*On the complexity of probabilistic image retrieva*l", in *ICCV'01*, Vancouver, 2001.

[39] C.M. Bishop, "*Neural Networks for pattern recognition*", Oxford, England: Oxford University Press, 1995.

[40] R.O. Duda, P.E. Hart, D.G. Stork, *"Pattern classification, second edition*", a Wiley-Interscience Publication, 2000.

[41] T. Kohonen, "*Self-organizing maps*", Springer-Verlag, Berlin, 1995.

[42] M. Dittenbach, D. Merkl, A. Rauber, *"The growing hierarchical self-organizing map*", in *Proc. of the Int. Joint Conf. on Neural Networks (IJCNN 2000)*, pages 15-19, Como, Italy, July 24-27 2000, IEEE Computer Society.

[43] F. Rousseaux, A. Bonardi, "*Reconcile art and culture on the web: lessen the importance of instantiation so creation can better be fiction*", in *Proc. First Int. Workshop on Philosophy and Informatics*, Cologne, Germany, 2004.

[44] C. Costa, J. Valle Jr., A. Koerich, "*Automatic classification of audio*", in *IEEE Int. Conf. on Systems, Man, Cybernetics*, The Hague, Netherlands 2004.

[45] T. Li, G. Tzanetakis, "*Factors in automatic musical genre classification of audio signals*", in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2003.

[46] C.J.C. Burges, *"A tutorial on support vector machines for pattern recognition*", in *Data Mining and Knowledge Discovery*, 2:121-167, 1998.

[47] J.L. Elman, "*Finding structure in time*", in *Cognitive Science*, vol. 14, pp. 179-211, 1990.

[48] L. Rabiner, "*A tutorial on Hidden Markov Models and selected applications in speech recognition*", in *Proc. of the IEEE*, 77:2 (257-286), 1989.

[49] N. Morgan, H. Bourlard, "*Continuous speech recognition: an introduction to the hybrid HMM/connectionist approach*", *in IEEE Signal Processing Magazine*, vol. 12, n°3, pp. 25-42, 1995.

[50] S. Katagiri, C.H. Lee, B.H. Juang, "*New discriminative training algorithms based on the generalized probabilistic descent method*", in *IEEE Proc. Workshop on Neural Networks for Signal Processing*, pp. 299-308, 1991.

[51] http://www.music-ir.org/evaluation/MIREX

7 Video Content Descriptors

7.1 State of the Art

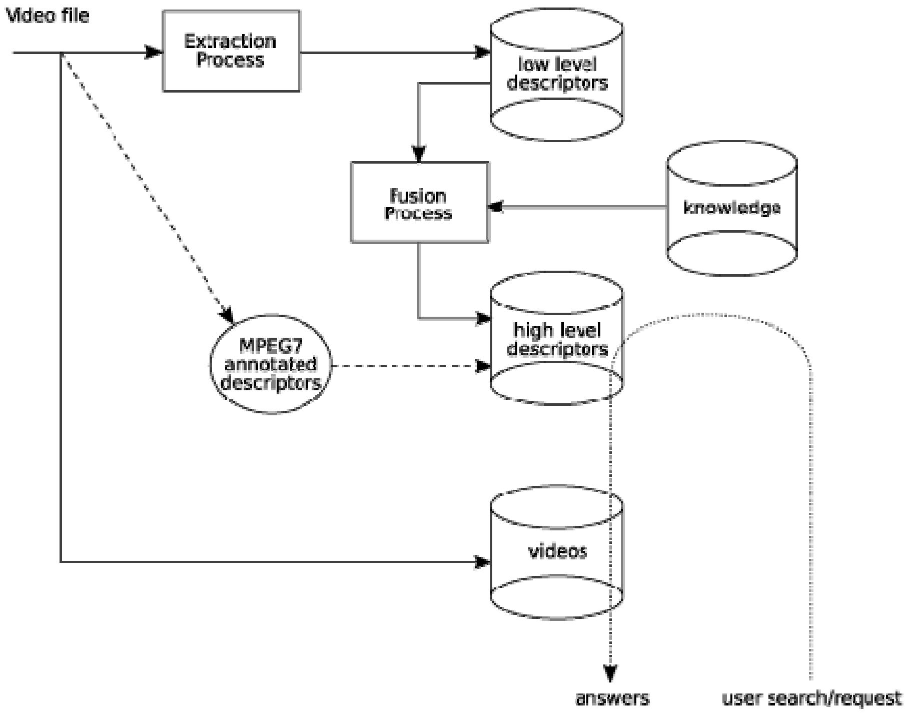
"An image is worth a thousand words", but video is a series of images, with sound in addition; therefore we should be able to collect a huge amount of metadata from the shortest video sequence. Linguistics-oriented considerations, like connotation and denotation are necessary in many cases. But we should remember that we can only achieve extraction of denoted semantics, and no connoted semantic (relative sense due to culture, context etc.), without a huge amount of knowledge and an appropriate artificial intelligence. It is well understood that the "weight" of the context, for example the cultural environment of the end user, can be decisive in the process of data indexing/storing/retrieving. A Video is an evolving 2-dimensional projection (evolving point of view, sequencing consists in cuts in the time-dimension) in a 4-dimensional physical environment with synchronised sound (1 dimension).

7.1.1 Structure of the extraction process

The structure of the extraction process is shown in the following figure. It can be seen that the low level descriptors are extracted from the video data. These low level descriptors can be combined with some kind of knowledge to result in high level descriptors – descriptors that are semantically meaning full for humans.

Examples for these high level descriptors are:

1. Voice recognition (who speaks) / extraction (what is said)
2. Music recognition (style) / extraction (notes)
3. Audio-video relations: global relations; local relations (explosions...)
4. face position, face recognition
5. Text extraction
6. events extraction (flashs...)



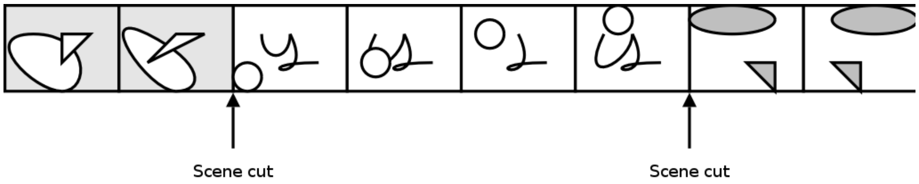
Low level descriptors are not semantically meaningful for humans. They can be calculated from

- inter-frame information, e.g. clip detection, moving blocks, …
- intra-frame information, e.g. dominant colours, textures, …

7.1.2 Low-level descriptors: Interframe

These descriptors are computed from the differences between frames, or from series of consecutive frames.

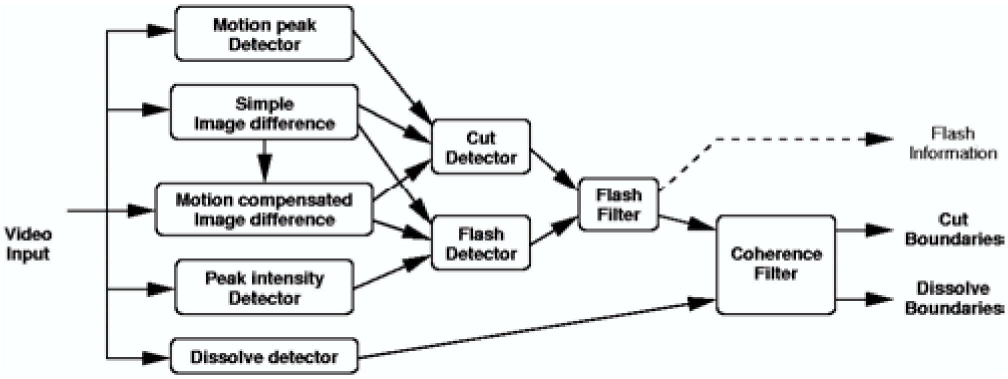
Scene Cut detection/sequencing/shot boundary detection/video parsing



Videos are mostly concatenations of different scenes, shot at various places or moments; scenes are the base elements of videos. Instead of processing the whole video, it can be cut into sequences, which have more or less a special content/meaning/interest at a semantic point of view. The descriptor is mainly a list of pointers to the detected scene cuts; statistics on length of scenes can be done.

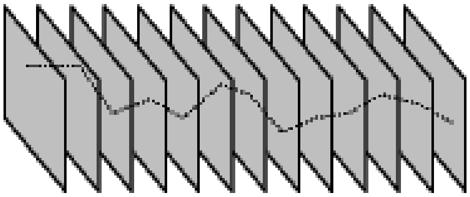
Comparison between consecutive frames can lead to false positives for cut detection (flashes, bad encoding, source chemical film degradation), therefore series of frames must be examined, corrections made to images, thresholding etc. in order to eliminate these artefacts. Moreover, a lot of inter-scenes are not pure cuts but transitions (like cross-fades, sweeps etc.) and more elaborate processes are necessary.

In [4]: The process has to find video sequence boundaries with (times-tamp, type of transition). They use the "CLIPS-IMAG shot segmentation system" to detect two types of transition effects: "cuts" and "dissolves" (including fades in and out) It detects cut transitions by direct images comparison after motion compensation and dissolve transitions by comparing the norms of the first and second temporal derivates of the images. It eliminates some artefacts thanks to a special module which detects photographic flashes and a special module consisting of a motion peak detector.



In [3]: evolution of grey-level centroids over time. They discuss about a relation between audio and visual signature. They show that audio-visual signature is more robust than video-only signature and permit better detection of video clips of shorter duration (about 2 seconds).

As they say, the video clip detection principle is the same for the audio, video or audio-visual signature. Over M frames, each of which an 8-dimensional vector is the signature, a distance is computed, on which an adaptive threshold is applied, function of the mean and standard deviation of the distance curve. They have chosen the grey level centroid as a visual signature, coordinates of which are computed as the weighted (by the grey value) sum of the pixel positions. This signature is valuable for a series of successive frames and characterizes spatio-temporal features. A single centroid can be insufficient, and applying it on sub-images like the four quarters is a good compromise. Then, an amplification of the centroids' movements can be done by biasing the computation toward brighter areas. The signature’s uniqueness is not certain but is a consequence of the complexity of the signature.

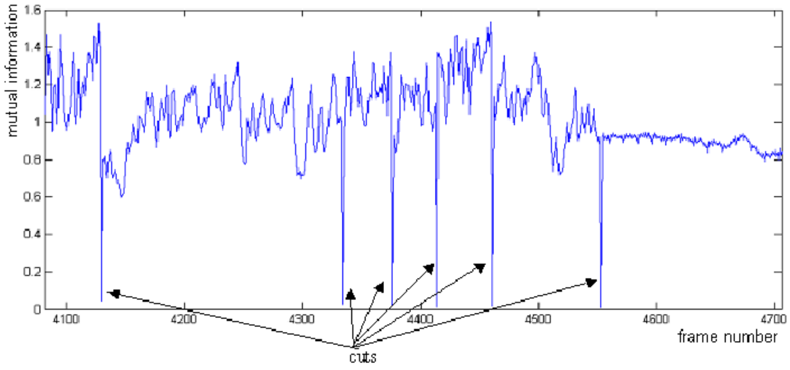


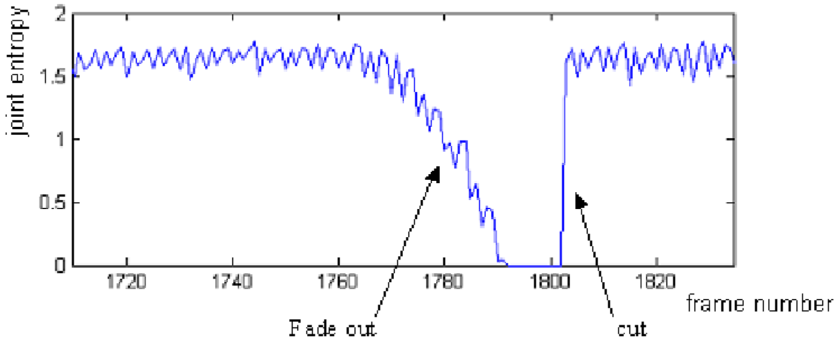
Note: this could build a video sequence signature, which could be useful in identifying sequences. It is very robust to very high compression ratio, with exactly the same results.

Others like [33] are based on the concept of continuity, extract scene boundaries. Uses the concepts of visual, position, camera focal distance, motion, audio and semantic continuity to group shots that exhibit some form of continuity into scenes. The TMR method (Chua et al 2000) filters commercials (Koh & Chua (2000)).

In [21] the detection and classification is based on dissolved video sequences. Multi-resolution of temporal slices extracted from 3D image volume.

The method proposed in [22] considers joint entropy between frames. CLIPS [19] experiments in Video Retrieval direct image comparison after motion compensation and "dissolve" transitions, comparing the norms of the first and second temporal derivates of the images. In [7] a quite simple cut detection algorithm is described. IRISA (http://www.irisa.fr/vista/Themes/Logiciel/MdShots/MdShots.english.html) proposed table of markers, each of which indicates a timestamp and a type of transition. It might also contain the duration of the transition.



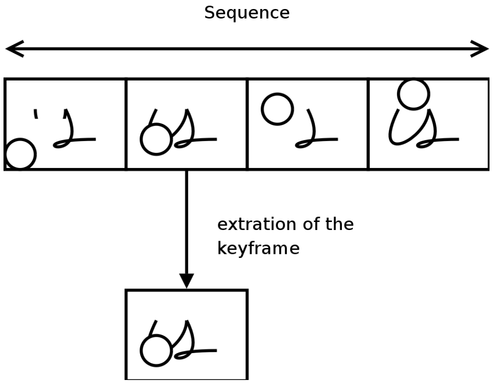


Visual perturbance

The visual perturbance characterises the amplitude of the motions inside a scene, at a global point of view, and over time. It is quite robust to different encodings of the same content and presents some semantic interest, possibly to classify the genre of a sequence, make searches, extract a (some) key frame(s). The descriptor is a couple of values evolving over time, corresponding to vertical and horizontal-direction perturbance.

Key framing - Extraction of a (the most) representative

For each sequence one or multiple frame can be chosen in order to be representative of the whole semantic content. It can be used for browsing in movies (iconographic overview), and also for some content extraction (which is quite limitative because this frame has no duration extend), or in order to recognize duplicates scenes in different movies.



In each sequence, the process extracts one or multiple frame which is/are identified as representative to the whole sequence. It could reveal unsuitable for identification of similar contents in case of low robustness. A simple and non-robust way to make this is to select the first frame of every scene; but it doesn't truly represent the real content of the scene, because of transitions, of complex action etc.

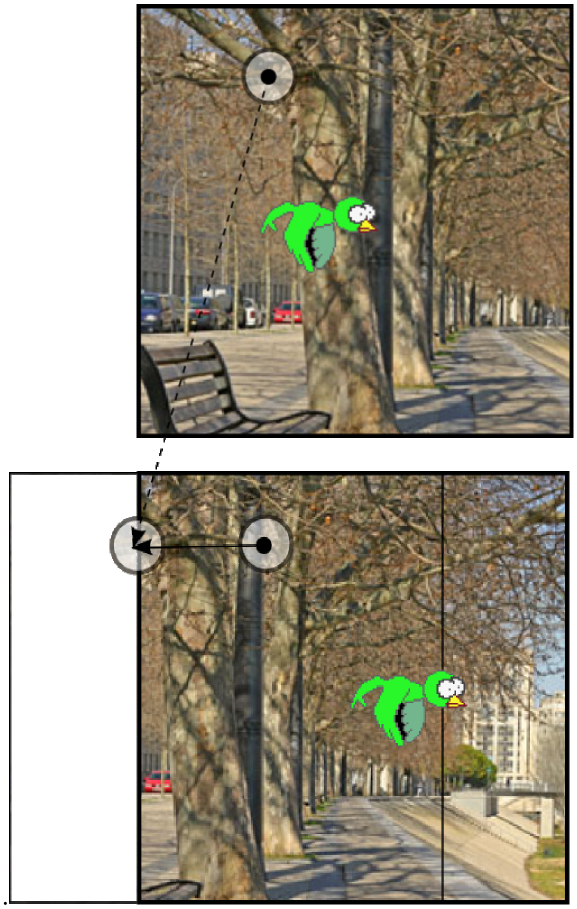
In [7] the Vitamin framework (a video content indexing system) is introduced: key frame detection, key frame signature extraction and key frame elimination technique, using DCT coefficients. The signature is build upon the DC and highest values of the AC coefficients of the MPEG or JPEG macro blocks. It acts at the middle of the compression/decompression process, making it available for real-time applications.

For key frame extraction, it assumes that first frame of a sequence is a key frame, and the first frame of a mostly static set of frames is a key frame; then it eliminates the repetitive and some meaningless frames (like a dialogue or black or unicolor screens). The robustness of this method is not proven.

The descriptor is a marker made of the timestamp of the key frame (or each key frame) and a pointer to the sequence (like a couple of sequence boundaries - i.e. detected transitions).

Background motion tracking & background extraction

When the video camera changes its viewing angle (for instance in order to make a panorama or to follow actors in a scene), the background acts like more or less like a still image translating and/or rotating, while some objects in the middle of the frame move.

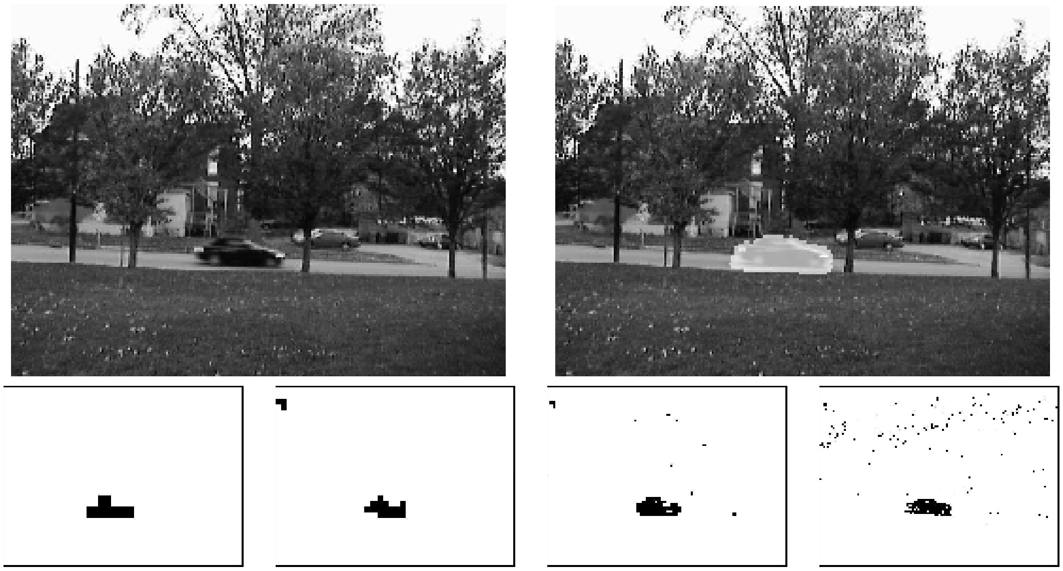


The background can reveal quite representative of the semantic surrounding of a scene (environment recognition/classification), while the camera motion can act like a perturbance value. It is quite robust to different encodings of the same content, and statistics over the camera movements can be useful for semantic identification and recognition of same content.

Reconstruct the motion of the background part of the sequence while eliminating objects of the foreground; possibly reconstruct the large background image. Lens deformations can perturb this process.

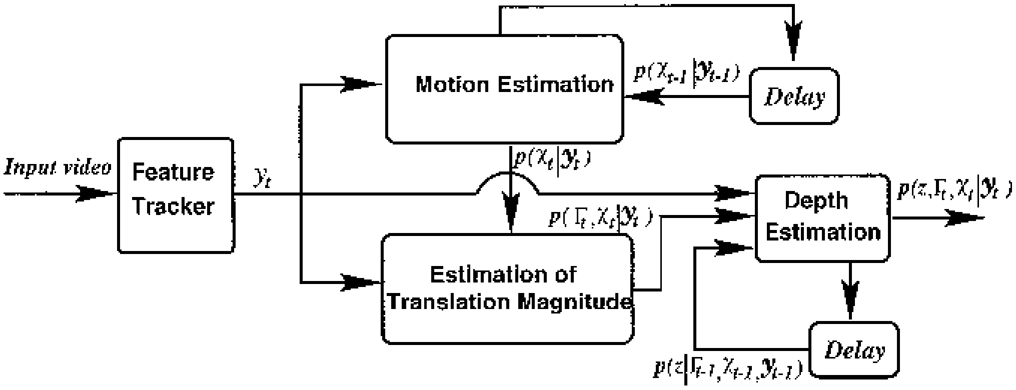
There are frameworks which work on still background, others which work on non-still backgrounds (in relief); the latter produces a 3-D reconstruction of the background and a detection of moving objects.

In [34] the frame work is based on background/foreground segmentation, construction of filters on sub-images, construction of a background model, multi-resolution approach. For each pixel and each resolution, spatial variation is calculated, in order to classify it. A hierarchical tree visit is made on the titles from the low resolution to higher resolution, in order to classify the block/have a more precise idea of the composition of the block. It has been well tested for diverse application purposes, false positives 8-12 % which is rather small.



The method proposed in [27] is based on random sampling, in order to obtain camera motion structure and scene structure (recover 3D from 2D). It is a relatively high-level process. There is a proposed framework with possible ameliorations.

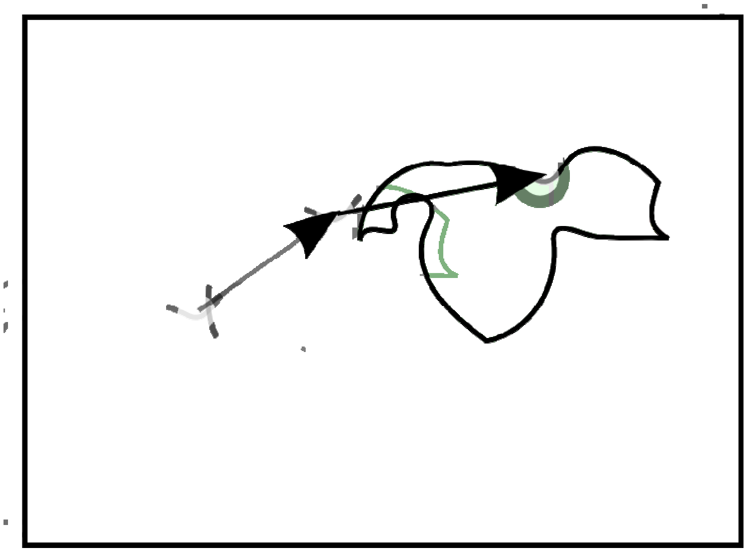
In [11] a method is proposed that permits to recover scene 3-D structure, moving objects (only linearly), and random camera motion. There is no evaluation of their results which seem qualitatively quite acceptable.


Input video
Feature Tracker
y_t
Motion Estimation
$p(\chi_{t-1}|\mathcal{Y}_{t-1})$
Delay
$p(\chi_t|\mathcal{Y}_t)$
$p(\Gamma_t,\chi_t|\mathcal{Y}_t)$
Estimation of Translation Magnitude
Depth Estimation
$p(z,\Gamma_t,\chi_t|\mathcal{Y}_t)$
Delay
$p(z|\Gamma_{t-1},\chi_{t-1},\mathcal{Y}_{t-1})$

Tensor voting based tracking is proposed in [14]; but it uses multiple stationary cameras, therefore it is bad within AXMEDIS. It needs motion blobs: background model with statistical techniques. The applicability depends on purposes (video surveillance etc.), but it seems to be applicable to a too tight domain. Other methods are presented in [32] or [6].

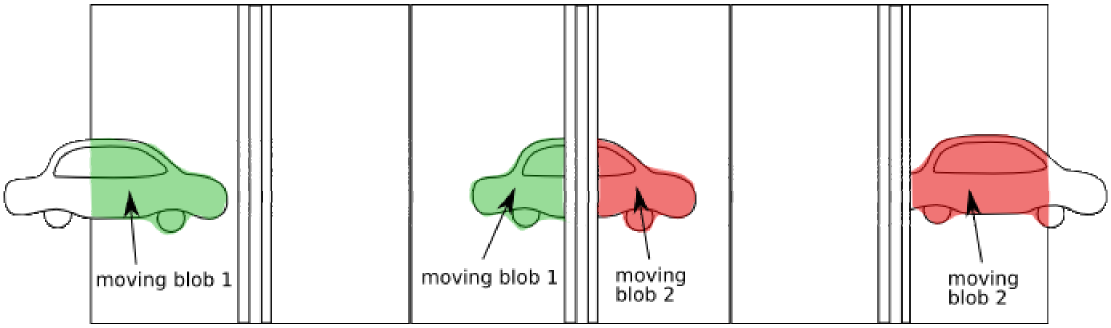
Block shape and motion tracking

Some portions ("blobs") of the image can move over the background. These blobs can have their shape and motion extracted.



At a semantic point of view, the shape, the dimension, the trajectory and relations between objects are meaningful. Statistics on size/movements and some high-level semantic purpose can use it, for instance for identifying the genre of a video film or following players in a sport video film.

An unsolved issue is the overlapping of objects: one object can overlap another object, making the former object totally disappear, at a computer point of view; then a new one (and one) appears. Then, two different objects can be recognized when only one has been hidden.



The descriptor is a set of curves (one per "blob") moving over time.

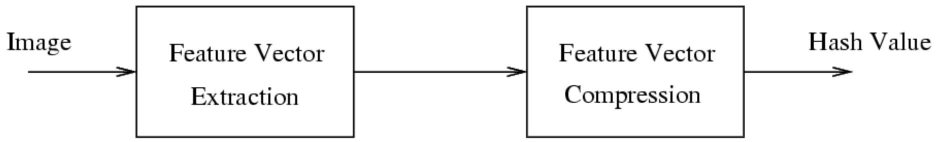
7.1.3 Interframe/intraframe

These descriptors can be applied either between frames or in some specific frames.

Image/video hash value

It's a one-way process which outputs a number (a short bit stream) from an frame (an image). If two images are perceived identical at a visual point of view (that means that the binary data have slightly different values), the output hash value should be the same (whereas binary data hash functions will output totally different numbers for two different images). It can be associated with a robust key framing process or be done on each frame.

It aims at simplifying the recognition of duplicate frames/sequences for instance in an image database: only the perceptual hash values need to be compared. The method proposed in [13] splits the decomposition of the hash process into two stages: the feature vector extraction and the hashing of this vector.

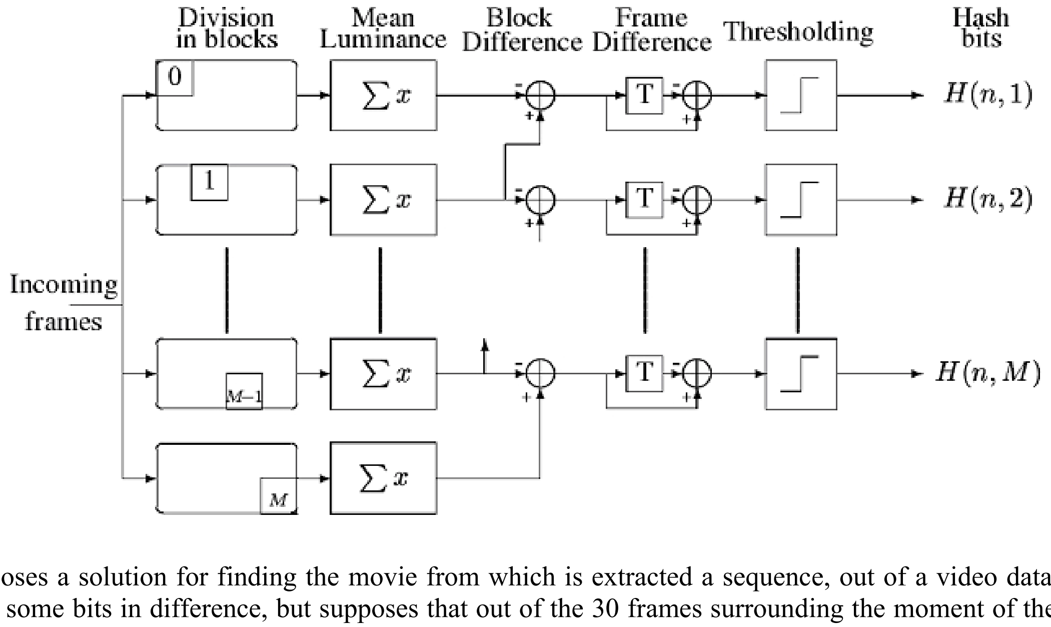


The features which can be extracted vary from filtering of the radon transform to extraction of feature points like used in watermarking or to statistics on sub bands of wavelet decomposition of parts of the image. Then, using a distributed coding method, it produces a hash value.

The experimental results seem quite good: few or no bit difference for most of the slight modifications of the image (1 degree rotation, scaling, flipping, 1% cropping, low pass filter, row/column deletion, linear transform), thus, being "robust", and bit difference between different images mostly around 50%. But too few statistics have been made on their results on order to verify the distribution of the number of bits in difference.

The separation into two stages raises an issue about the utility of the second part of the process: the vector extraction, if robust, would satisfy the application purposes in case only of detection of copy of an image in a database; the encoding of the vector is mostly necessary in order to prevent an attacker to make an image have the same vector as the vector he has already seen in a database. This latter condition is very tight in case of a close database. Anyway, it seems that later encryption (or hashing) of the extracted feature vector would lead to the same advantages as the ones of the process described above.

Monga et al. [2] present a clustering algorithm for the hashing of the feature vector. In [23] a robust video sequence hash process is presented, which is not any more a "hash" in the cryptographic sense because visual differences don't lead to half of the bits switched but to small changes.



It proposes a solution for finding the movie from which is extracted a sequence, out of a video database. It admits some bits in difference, but supposes that out of the 30 frames surrounding the moment of the movie for which we expect to find a duplicate in our movie database, there will be one (frame) which will have no bits in difference. Then, it tries to match the hash values of sequences found in the database and the current sequence; it evaluates the Hamming distance on the 30 frames, thresholds it and answers.

The authors in [30] say that global DCT is more robust to compression than wavelets and local block-based DCT. Descriptor Set of bits associated with a timestamp (frame pointer) or a block (group of frames) timestamp.

Image/video segmentation

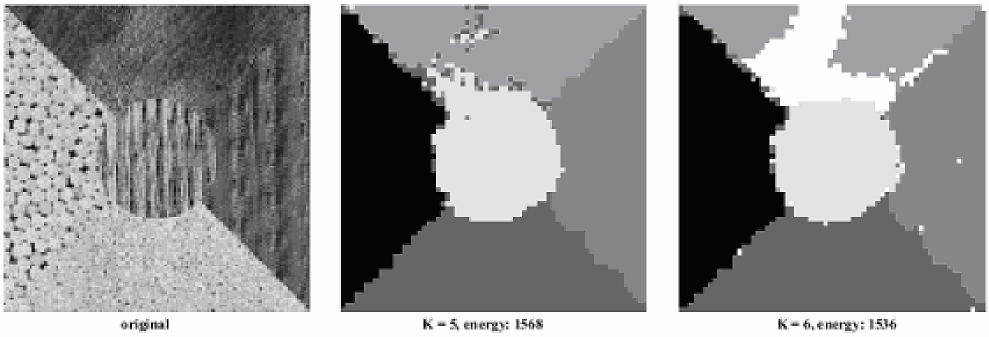
In each frame, the image is perceived by the human eye as segmented into regions with a certain unicity of colour, texture, etc. The image segmentation process identifies regions and contours in each image (or one image out of a certain number) or key frame, delimiting homogenous colour/texture areas, possibly their evolutions through the video sequence.

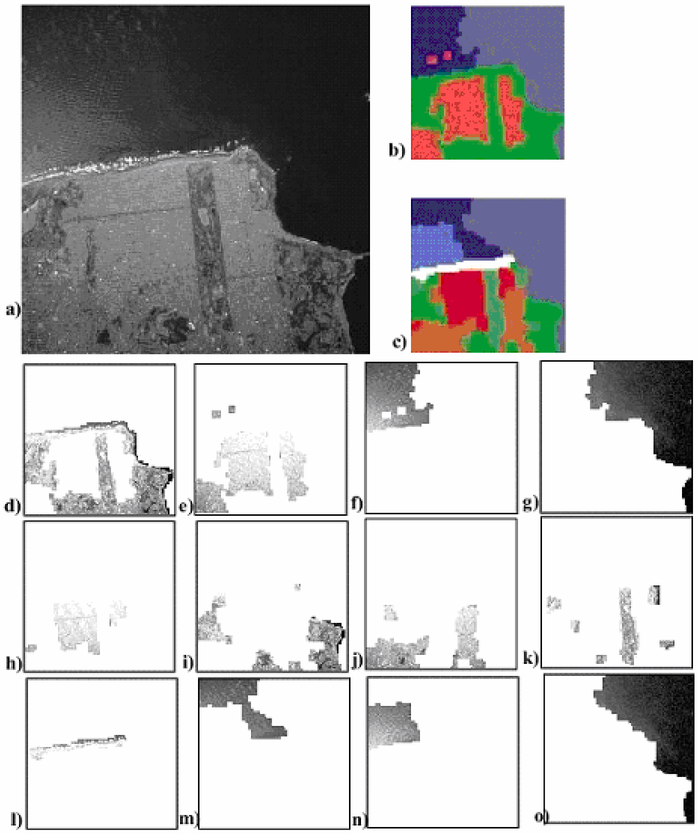
Give the shape of objects or sub-objects and their content (colour/texture) characteristics in order to make statistics or allow searches in a database of these shapes and/or colour/texture characteristics

How to "optimally partition an image into homogeneous regions" [26]. It has been well-tested on still image. What is more undetermined is when it is evolving over time; succeed in recognising moving contours of same-content zones, while the content characteristics are evolving.

In [16] geodesic active contours and edge flow segmentation are used for a semi-automated segmentation method. It works on intensity, colour, and texture features. An initial contour is deformed towards the object boundary (see also [17] and [18]). The method proposed in [5] is quite old and uses mean shift algorithm.

In [26] texture segmentation (quite old: 1998) "unsupervised segmentation" is proposed. It says that a general vector space approach has huge drawbacks because the used metric is adapted to a too small population of image content. They use instead non-parametric statistical tests, which they find reliable to measure local texture similarities. The method proposed in [29] is based on Gabor decomposition, vector space approach, and auto classification.





Further algorithms based on segmentation can be found in [15], [28] and [26]

Texture hash value

A texture hash method is a one-way process which outputs a more or less unique number from a texture in a certain area of an image. It is very similar to the image segmentation and the image hash function processes. It can be used to recognise some elements appearing in different scenes. Depending on the process, we can extract two types of

- Intra frame (still shape): key frame pointer, shape, texture hash value
- Inter frame (evolving shape): sequence pointer, shape evolving over time, texture hash value

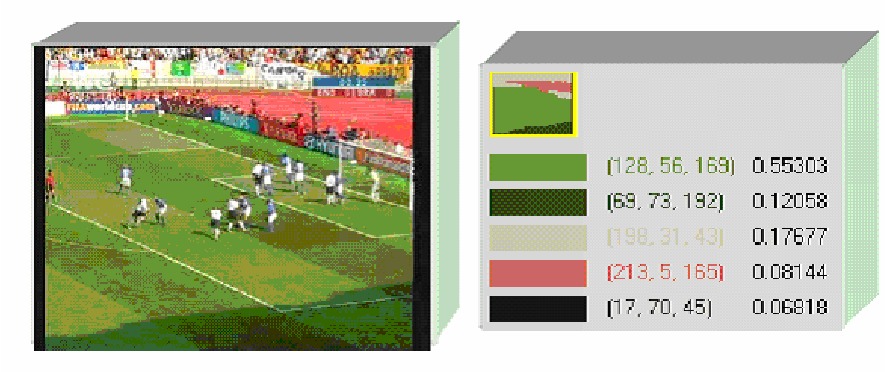
7.1.4 Intraframe

These descriptors are computed from a single frame.

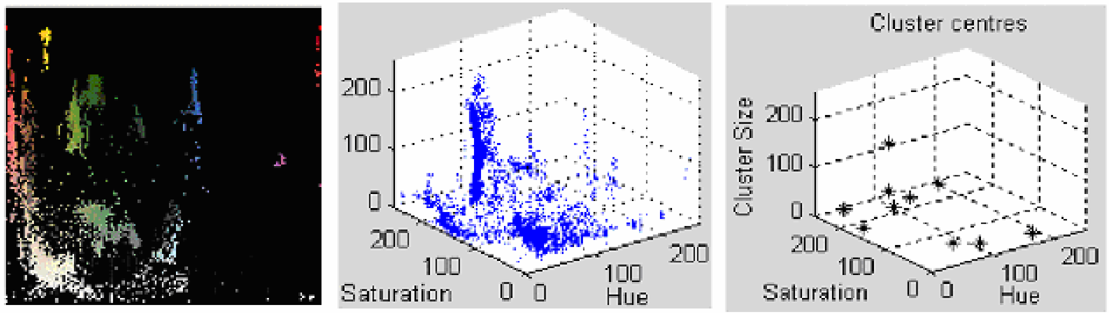
Colour space

The first, second and third colour moments of spatial sub regions of the video sequence can be extracted and computed, the colour quantization of the container can also be extracted. Their histogram and evolution along time give some semantic information about the content (for classification etc.) The robustness of the histogram and its spatio-temporal evolution is quite relative: it is not robust to analog copies, some types of re-encodings, but it is robust to scaling and rotation of the video, and content variations like slightly different points of view, and some occlusions (see also [8]).

In [8] the authors say that the histogram is not robust and effective for colour characterization (e.g. dominant colour) in large video databases, because of the lack of context and spatial information. It presents nonparametric colour characterisation model (NCCM) using mean shift procedure, with an emphasis on spatio-temporal consistency. The process selects the mean colour sets of a sequence (thanks to a good colour mode seeking function), through consecutive frames, and computes their extend over the screen.



🟩	(128, 56, 169)	0.55303
▨	(69, 73, 192)	0.12058
⬜	(198, 31, 43)	0.17677
🟥	(213, 5, 165)	0.08144
⬛	(17, 70, 45)	0.06818



Some specific applications of this process (e.g. detection of a tennis court scene in a sport sequence) have very good results (e.g. recall 98.7%, precision 97.2%). The process is quite fast and permits (more than) real-time applications.

7.2 Problems

The problems of the individual low-level descriptors are identified and described in the previous section ‘state of the art’ in the description of the individual low-level descriptors.

Besides the problems described in the previous section of the individual low-level descriptors, one general problem can be identified: High-level descriptors are only available for specific types of video sequences or specific scenarios. For example for the surveillance of high ways the position of the cars and lorries can be extracted.

7.3 Work Performed

- Analysis state of the art: The state of the art has been analysed. The focus was on low-level descriptors and existing frameworks. Existing frameworks suggest the combination of different descriptors. Depending on the descriptor the corresponding database has to be designed.

- Evaluation criteria have been identified. Different methods for benchmarking and evaluating video descriptors and their extraction algorithms have been published. These methods have been analysed and an evaluation scheme suitable for AXMEDIS was identified.

- The direct result of the state of the art analysis and the identified evaluation criteria was the pre-selection of suitable video descriptors.

7.4 References

[1] Netra V spatio-temporal segmentation examples, http://vision.ece.ucsb.edu/netra/examples/netrav.html

[2] Arindam Banerjee, Brian L. Evans, and Vishal Monga. Clustering algorithms for perceptual image hashing, June 13 2004.

[3] Laurent Besacier Isabelle Simand Stphane Brs Benjamin Senechal, Denis Pellerin. Audio, video and audio-visual signatures for short video clip detection: Experiments on trecvid2003. 2005.

[4] Laurent Besacier, Georges Qunot, Stphane Ayache, and Daniel Moraru. Video story segmentation with multi-modal features: experiments on trecvid 2003. In MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval, pages 221-227, New York, NY, USA, 2004. ACM Press.

[5] Dorin Comaniciu and Peter Meer. Robust analysis of feature spaces: Color image segmentation, November 22 1997.

[6] Nevenka Dimitrova and Forouzan Golshani. Motion recovery for video content classi_cation. ACM Transactions on Information Systems, 13(4):408-439, 1995.

[7] Nevenka Dimitrova, Thomas McGee, and Herman Elenbaas. Video key frame extraction and _ltering: A key frame is not a key frame to everyone. In Forouzan Golshani and Kia Makki, editors, Proceedings of the 6th International Conference on Information and Knowledge Management (CIKM-97), pages 113-120, New York, November 10-14 1997. ACM Press.

[8] Ling-Yu Duan, Min Xu, Qi Tian, and Chang sheng Xu. Nonparametric color characterization using mean shift. In Proceedings of the 11th ACM International Conference on Multimedia (MM-03), pages 243-246, November 4-6.

[9] B. Frey, M. R. Naphade, T. Kristjansson, and T. S. Huang. Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems., July 22 1998. vol. 3, pp. 536-540.

[10] Arun Hampapur and Ruud Bolle. Videogrep: Video copy detection using inverted _le indices. Technical report, IBM Research.

[11] Mei Han and Takeo Kanade. Reconstruction of a scene with multiple linearly moving objects, June 27 2000.

[12] ISO. Mpeg-7 overview, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

[13] Mark Johnson and Kannan Ramchandran. Dither-based secure image hashing using distributed coding. In ICIP (2), pages 751-754, 2003.

[14] Jinman Kang, Isaac Cohen, and Gerard Medioni. Multi-views tracking within and across uncalibrated camera streams.

[15] Iasonas Kokkinos, Georgios Evangelopoulos, and Petros Maragos. Modulation-feature based textured image segmentation using curve evolution. In ICIP, pages 1201-1204, 2004.

[16] B. S. Manjunath, Baris Sumengen, and Charles Kenney. Image segmentation using curve evolution, March 27 2002.

[17] B. S. Manjunath, Baris Sumengen, and Charles Kenney. Image segmentation using curve evolution and owelds, July 11 2002.

[18] B. S. Manjunath, Baris Sumengen, and Charles Kenney. Image segmentation using curve evolution and region stability, June 10 2002.

[19] Daniel Moraru, Georges M. Quenot, and Laurent Besacier. CLIPS at TREC-11: Experiments in video retrieval, February 06 2003.

[20] Xiangming Mu and Gary Marchionini. Statistical visual feature indexes in video retrieval. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Posters, pages 395-396, 2003.

[21] Chong-Wah Ngo. A robust dissolve detector by support vector machine. In Proceedings of the 11th ACM International Conference on Multimedia (MM-03), pages 283-286, November 4-6.

[22] Cernekov C. Nikou and I. Pitas. Shot detection in video sequences using entropy-based metrics, October 02 2002.

[23] J. Oostveen, Kalker A.A.C.M., and J. Haitsma. Visual hashing of digital video: applications and techniques. In Proc. SPIE; Applications of Digital Image Processing XXIV. 4472, pages 121-131, 2001.

[24] Milan Petkovic and Willem Jonker. Content-Based Video Retrieval : A Database Perspective (Multimedia Systems and Applications). Springer, October 2003.

[25] Kok Meng Pua. Feature-based Video Sequence Identi_cation. PhD thesis, the Univ. of Kansas, 2002.

[26] Jan Puzicha, Joachim M. Buhmann, Rheinische Friedrich wilhelms universitat, and Thomas Hofmann. Unsupervised texture segmentation in a deterministic annealing framework, May 22 1998.

[27] Gang Qian and Rama Chellappa. Structure from motion using sequential monte carlo methods. In Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), pages 614-621, Los Alamitos, CA, July 9-12 2001. IEEE Computer Society.

[28] Mikal Rousson, Projet Odyssee, Rachid Deriche, and Thomas Brox. Active unsupervised texture segmentation on a diffusion based feature space. Technical report, February 25 2003.

[29] Jianbo Shi, Jitendra Malik, Serge Belongie, and Thomas Leung. Contour and texture analysis for image segmentation, 2001.

[30] Champskud J. Skrepth and Andreas Uhl. Robust hash functions for visual data: An experimental comparison. In IbPRIA, pages 986-993, 2003.

[31] Alan F. Smeaton, Wessel Kraaij, and Paul Over. The TREC Video retrieval evaluation (TRECVID): A case study and status report, dec 2003.

[32] Sundar Vedula, Simon Baker, and Takeo Kanade. Image-based spatio-temporal modeling and view interpolation of dynamic events. ACM Transactions on Graphics, 24(2):240-261, April 2005.

[33] JihuaWang and Tat-Seng Chua. A framework for video scene boundary detection. In Proceedings of the tenth ACM international conference on Multimedia (MM-02), pages 243-246, New York, December 1-6 2002. ACM Press.

[34] Quanren Xiong and Christopher Jaynes. Multi-resolution background modeling of dynamic scenes using weighted match _lters. In VSSN '04: Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks, pages 88-96, New York, NY, USA, 2004. ACM Press.

[35] Xianfeng Yang, Qibin Sun, and Qi Tian. Content-based video identification: A survey.

8 Text Documents Content Descriptors

High-level metadata are needed for enhanced document retrieval in a multimedia environment. Information about the keywords and the domain of a document allows advanced searches within the collection of documents, improving the standard information retrieval techniques with a set of descriptors which represent the contents.

The procedure outlined in this section aims to exploit current resources and tools and to produce an integrated architecture for keyword (and domain) extraction. In this perspective, it is important to point out the need for an open-domain extraction technique, i.e. a technique that is not restricted to a specific type of document. A further task to be taken into account is the multilingual environment of the repositories. This is relevant in multimedia that may comprise all kinds of contents in many languages.

The proposed system tries to respond to both requirements, integrating the process of keyword extraction with multilingual resources which allow the treatment of documents written in different languages (covering English, French, German, Italian, and Spanish), regardless of their domain and typology.

8.1 State of the Art

Previous experiences of keyword detection procedures pointed out two main strategies, that have been merged in various ways according to practical needs: (a) the use of statistical-based technologies; (b) the use of linguistic-driven tools and databases.

The statistical-based technologies mostly rely on **statistical comparison** with a general corpus. For example, Salton [9] suggested the TF.IDF to capture the “weight” of a word in a collection of documents (using the words frequency of distribution). Other well-known proposed methods are Mutual Information (MI), by Church and Hanks [2]; log-likelihood measure, by Dunning [4]; χ^2 measure[1].

Exploiting the last of these measures, Matsuo [8] proposed a keyword extraction algorithm based on the statistical analysis of a single document, starting from words-association measures of co-occurrence in a given context (i.e. the period). Such a technique is particularly suitable for languages with a lack of resources (PoS taggers, referring corpora etc.), since it doesn’t require any further tool but a simple stemmer.

In general, linguistic-driven techniques start to run after a first statistical analysis. For example, in Van der Plas et al. [13] there is a first stage of statistical analysis based on RFR (Relative Frequency Ratio) algorithm, and a second stage of semantic analysis, based on lexical databases such as WordNet (see below) or EDR (an electronic dictionary).

Within the JRC experience (see [12]; http://www.jrc.cec.eu.int/langtech/index.html) it was proposed to merge different methods in order to construct a two-stages architecture for keyword extraction in a

[1] See Kilgarriff [6] and Chung [1] for the evaluation of the results achieved with different techniques.

multilingual environment. As a first step, the JRC method assigns to a given document an open-set of monolingual keywords, using a statistic method based on the log-likelihood algorithm. The monolingual keywords are then associated to domain-descriptors belonging to a restricted and pre-defined set, the EUROVOC thesaurus, “a multilingual thesaurus covering the fields in which the European Communities are active; it provides a means of indexing the documents in the documentation systems of the European institutions and of their users” (http://europa.eu.int/celex/eurovoc/). This parallel multilingual resource assures the translatability of the information extracted.
The IRC experience points out strategies that may also be significant in a multimedia environment, however the use of thesauruses cannot be maintained once the open-domain requirement is taken into account. The following sections will outline three alternative algorithms needed to ensure metadata extraction in open-domains, combining statistic methods and available language resources and tools.

Proposal based on the state of the art

The principles of the three main algorithms (Algorithm A, Algorithm B, and Algorithm C) that will be used for keyword-extraction procedure are described as follows. The description consists of two parts:

1. Pre-processing: Step 1 – 3
2. Content Description/Meta data Extraction: Step 4 and following

To extract metadata and content descriptors, the following pre-processing steps has to be performed before extraction:

1. Conversion of the file from the given format to plain text format without representation or formatting information (e.g. ASCII-TXT).

2. Identification of the main language of the document through the statistical language guesser component. This is done by a statistical comparison between document characters n-grams distribution and reference statistics. The range of identifiable “language” metadata is English (en), French (fr), German (de), Italien (it), and Spanish (es).

3. Pre-processing

- *Disambiguation of the periods:* The identification of periods that have a full stop value is needed to tokenize the document in relevant linguistic units.
- *Text tokenization*: Different units of tokenization: words, chunks, periods.
- *Lemmatization:* Each word receives a tag regarding the part of speech (PoS) and the lemma. To be accomplished by an external multilingual tagging tool (e.g. TreeTagger, Xerox Tagger, Connexor Machinese) for de, en, fr, it.
- *Selection of nouns*: From the results of lemmatization, only nouns are assumed to be potential keyword/descriptors of the document. Output of the document-specific nouns frequency list.

Algorithm A

4A. Statistic Processing

- *Selection of the n% most frequent nouns*
- *Statistic clustering.* The clustering process is based on two co-occurrence measures of the output items of the previous step (co-occurrence is estimated within period contexts): similarity-based clustering (terms with similar distribution of co-occurrence with other terms even if they are not frequent, measured by the Jensen-Shannon divergence); pairwise clustering (terms that co-occur frequently in the same period contexts, measured by mutual information score)
- *Keyness value estimation.* Estimation of the importance of the pre-selected words of a document. Various algorithms can be used: χ^2, log-likelihood, z-score. The estimation generates a score for the keyword candidates.

5A. Fine grained collocation extraction.
Analysis of the strict co-occurrence (in contiguous context of fixed length) of each selected keyword with other related words. If the co-occurrence overruns a fixed threshold (to be experimentally detected) the keyword become a complex keyword.

6A. Output of the document keyword
Selection of the words (or multi-words) with the highest score.

Algorithm B

4B. Statistic comparison with general corpus frequency lists.

The frequency list extracted from a specific document is compared with a general one, extracted form a big corpus.The more prominent differences (mostly within the nouns) are taken into account as a fist step to recognize the document specific lexicon, that is assumed to contain the relevant keyword candidates. The TFIDF algoritm has been choose to accomplish this task (In principle the result of this process could be also compared , and refined, with the internal statistic scoring discussed at point 4A.)

5B. Fine grained collocation extraction.

Analysis of the strict co-occurrence (in contiguous context of fixed length) of each selected keyword with other related words. If the co-occurrence overruns a fixed threshold (to be experimentally detected) the keyword become a complex keyword.

6B. Output of the document keyword

Selection of the words (or multi-words) with the highest score(s).

Algorithm C

4C. Statistic comparison with general corpus frequency lists.

The frequency list extracted from a specific document is compared with a general one, extracted form a reference corpus. The more prominent differences (mostly within the nouns) are taken into account as a fist step to recognize the document specific lexicon, that is assumed to contain the relevant keyword candidates. The result of this process can be compared (and refined) with the internal statistic scoring discussed at point 4A.

5C. Output of the keyword candidates.

Wide selection of the words (or multi-words) with the highest score(s).

6C. Semantic processing

- *Word Sense Disambiguation (WSD):* Words can be ambiguous about their meaning. With respect to this point, in WordNet a word can be associated with one or more concepts, that describe the possible meanings of a single lemma. WSD strategies take into account the semantic context in which the word occurs. The output is the association of each word with one concept, that makes possible the translation of the selected (key)words into the different languages (associated in the multilingual WordNet).
- *Semantic Similarity estimation (SS):* The Leacock-Chorodow concept similarity measure can be used to measure the semantic similarity between two concepts. This measure is based on the length of the paths between pairs of concepts in the WordNet IS-A hierarchy.
- *Semantic clustering:* Clusters of concepts related to each other with decreasing values of *SS* (up to a predefined lower threshold).
- *Scoring:*
    1. first level: *cluster level score*: This score is defined as the sum of two components:
        - *connectivity*, corresponding to a global measure of the semantic relatedness between all concepts in the cluster;
        - *reiteration*, corresponding to a global measure of the importance of the concepts in the cluster.
    2. second level: *concept level score:* This score is defined in a similar way as the sum of two components:
        - *concept level connectivity*;
        - *concept level reiteration*.

            These two measures are combined by a function that produces a unique score for each concept.

            Words that correspond to concepts with the highest global score are the best candidates to be document keywords.

            The cluster with the higher score can be associated with a descriptor (as the ones in the MultiWordNet Domains). The result is the output of the domain of the document.

7C. Fine grained collocation extraction.

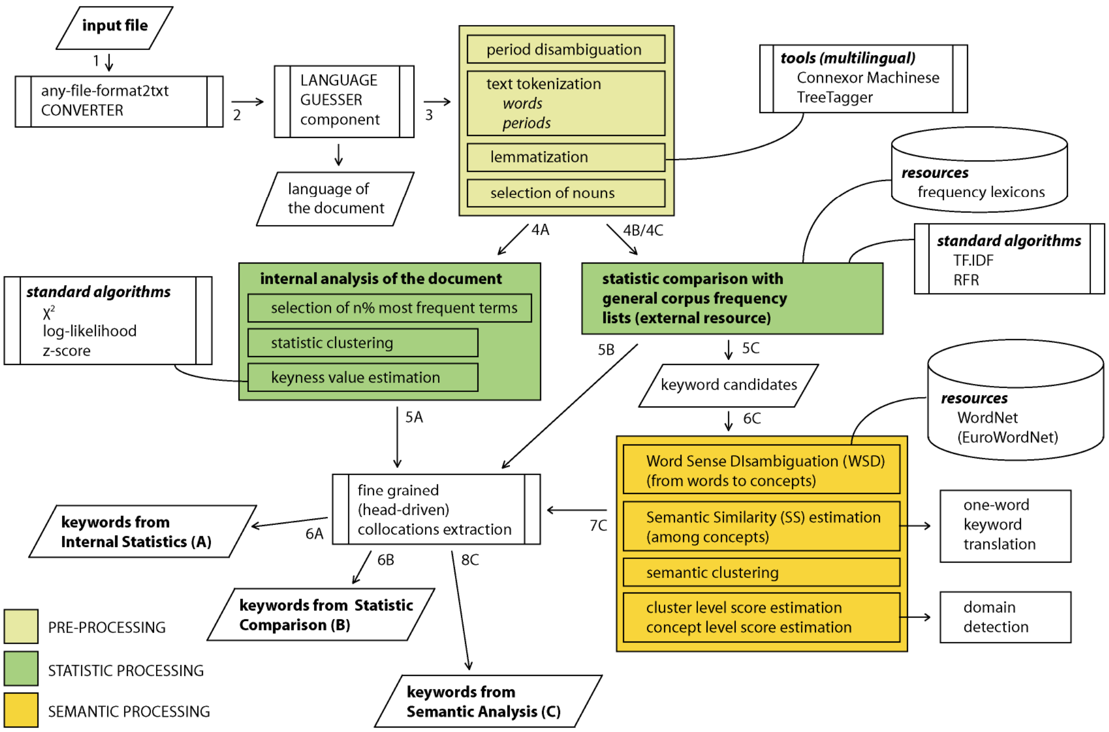
Analysis of the strict co-occurrence (in contiguous context of fixed length) of each selected keyword with other related words. If the co-occurrence overruns a fixed threshold (to be experimentally detected) the keyword become a complex keyword.

8C. Output of the document keyword

Words (or multi-words) with the highest score(s) are selected and returned.

The components needed for the extraction of metadata from text documents and the information flow in between them is shown in the next figure.

LANGUAGE INFRASTRUCTURE FOR METADATA EXTRACTION IN OPEN-DOMAIN MULTILINGUAL TEXTUAL RESOURCES


input file
1
any-file-format2txt CONVERTER
2
LANGUAGE GUESSER component
language of the document
3
period disambiguation
text tokenization
words
periods
lemmatization
selection of nouns
tools (multilingual)
Connexor Machinese
TreeTagger
resources
frequency lexicons
4A
4B/4C
standard algorithms
TF.IDF
RFR
standard algorithms
χ²
log-likelihood
z-score
internal analysis of the document
selection of n% most frequent terms
statistic clustering
keyness value estimation
statistic comparison with general corpus frequency lists (external resource)
5B
5C
keyword candidates
6C
resources
WordNet
(EuroWordNet)
5A
fine grained (head-driven) collocations extraction
keywords from Internal Statistics (A)
6A
7C
Word Sense DIsambiguation (WSD) (from words to concepts)
Semantic Similarity (SS) estimation (among concepts)
semantic clustering
cluster level score estimation
concept level score estimation
one-word keyword translation
domain detection
6B
8C
keywords from Statistic Comparison (B)
keywords from Semantic Analysis (C)
PRE-PROCESSING
STATISTIC PROCESSING
SEMANTIC PROCESSING

8.2 Problems

1. Reference statistics for all languages other then English are not freely available. **Solution:** Statistics will be implemented by DIPITA.

2. Neither Multilingual tagger nor Language infrastructures like WordNet are freely avaliable for commercial use. The Use of Tree-Tagger and WordNet -English are free for research, but they subjet to licence for commercial use.
Solution: To be provided in accordance with the AXMEDIS general policy

3. The Use of WordNet for *Semantic Similarity estimation (SS) and Semantic clustering*, that have been proposed in the literature, leads partially to negative results, due to the internal structure of the data base. It turns out that Semantic similarity estimation based on the distance between Synsets can be used for word sense disambiguation but not for clustering.
Solution: A version of Word-net for English where synsets are linked to Domain information is available for research (Wordnet Domains 3.0 http://tcc.itc.it/research/textec/topics/disambiguation/download.html).

4. Translation of text descriptors to MPEG-7 format is problematic due to the lack of specific descriptors for text in the MPEG standard, which is oriented to the multimedia objects. More specifically, so far text is considered an attribute of the object and not the content of the object. As a consequence descriptors for text characteristics are not taken into account.
Solution: A proposal to MPEG-7 standard body for the integration of text descriptors will be presented.The proposal will be based on the existing standard for text description (e.g.: Text Encoding Initiative)

8.3 Work Performed

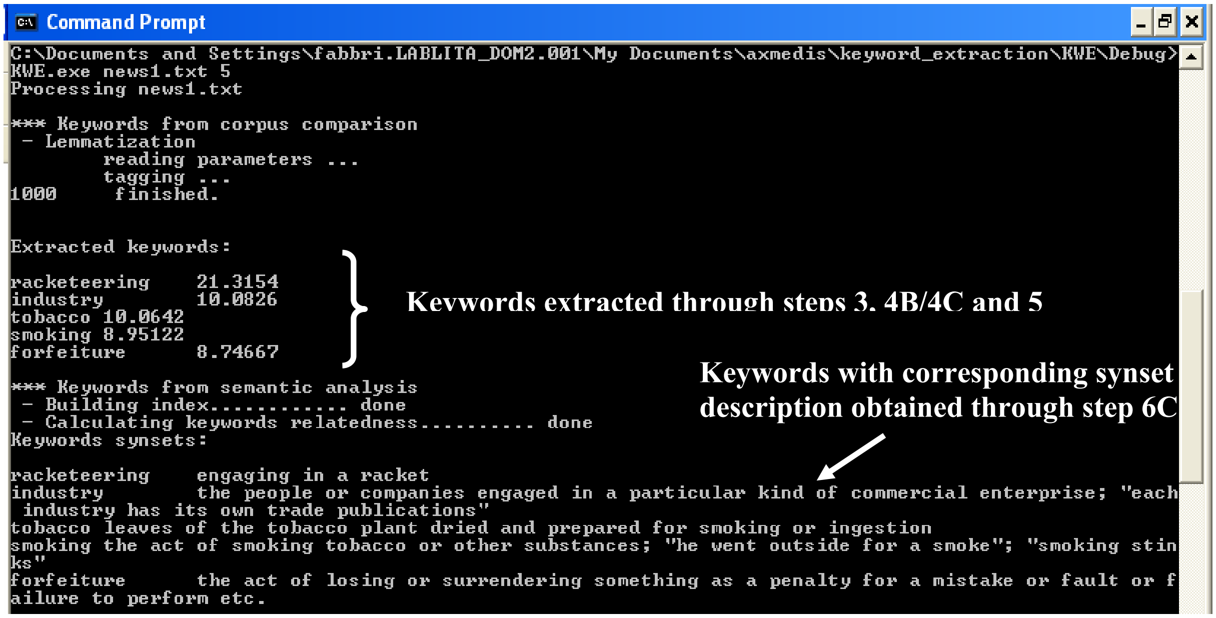
1- Language resource collection:
a) Multilingual Taggers; b) Reference Corpus for English; c)WordNet Data Base for English

2- Collection of Parallel texts (en; it, fr, sp, ge) for testing (see DE8.1.1)

3- A prototype demonstrator working on Microsoft Windows platform has been posted in the AXMEDIS' CVS repository. The demonstrator performs the following core components of Keyword extraction, limited to English documents (numbered in accordance with the above description):

3. Pre-processing. a) Text tokenization; b) Lemmatization (accomplished with TreeTagger)
Selection of nouns
4B/4C. Statistic comparison with general corpus frequency lists.
The frequency list extracted from a specific English document is compared through TFIDF algoritm with the statistics generated from the British National Corpus, taken as general reference corpus .
5.C. Keyword candidates are extracted
6C. Semantic processing
Word Sense Disambiguation (WSD*) w*ith respect to WordNet data base for English
The WordNet synset description derived from the word sense disambiguation has been associated to each keyword candidate.

All components have been object of positive early testing by DIPITA
Below a snapshot of the output of the prototype demonstrator:



```
C:\Documents and Settings\fabbri.LABLITA_DOM2.001\My Documents\axmedis\keyword_extraction\KWE\Debug>
KWE.exe news1.txt 5
Processing news1.txt

*** Keywords from corpus comparison
 - Lemmatization
        reading parameters ...
        tagging ...
1000     finished.


Extracted keywords:

racketeering    21.3154
industry        10.0826
tobacco 10.0642
smoking 8.95122
forfeiture      8.74667

*** Keywords from semantic analysis
 - Building index............ done
 - Calculating keywords relatedness.......... done
Keywords synsets:

racketeering    engaging in a racket
industry        the people or companies engaged in a particular kind of commercial enterprise; "each
 industry has its own trade publications"
tobacco leaves of the tobacco plant dried and prepared for smoking or ingestion
smoking the act of smoking tobacco or other substances; "he went outside for a smoke"; "smoking stin
ks"
forfeiture      the act of losing or surrendering something as a penalty for a mistake or fault or f
ailure to perform etc.
```

Keywords extracted through steps 3, 4B/4C and 5

Keywords with corresponding synset description obtained through step 6C

news1.txt is an English plain text file containing a news regarding a civil case against some tobacco industries accused of fraud and racketeering; in this example, *5* keywords were requested.
Steps 3, 4B/4C and 5 extract keyword candidates (*racketeering*, *industry*, *tobacco*, *smoking* and *forfeiture*).
Step 6C disambiguates keywords candidates senses and associates the synset description to each candidate in accordance to WordNet synsets database.

8.4 References

[1] T. M. Chung , “A Corpus Comparison Approach for Terminology Extraction”, *Terminology*, 9(2), Benjamins, Amsterdam, 2003, pp. 221-246.

[2] K. W. Church, and P. Hanks, “Word association norms, mutual information, and lexicography”, *Computational Linguistics*, 16(1), MIT Press, Cambridge, MA, 1990, pp. 22-29.

[3] P. Drouin, “Term Extraction Using Non-technical Corpora as a Point of Leverage”, *Terminology*, 9(1), Benjamins, Amsterdam, 2003, pp. 99-115.

[4] T. Dunning,. “Accurate methods for the statistics of surprise and coincidence”, *Computational Linguistics*, 19(1), MIT Press, Cambridge, MA, 1993, pp. 61-74.

[5] Fellbaum, C. (ed.),. *WordNet: An electronic lexical database. MIT Press*, Cambridge, MA, 1998.

[6] A. Kilgarriff,, “Comparing corpora”, *International Journal of Corpus Linguistics*, 6(1), Benjamins, Amsterdam, 2001, pp. 1-37.

[7] C. Leacock, and M. Chodorow, “Combining local context and WordNet similarity for word sense identification”, in Fellbaum, C. (ed.), *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA, 1998.

[8] Y. Matsuo, and M. Ishizuka, “Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information”, *International Journal on Artificial Intelligence Tools (IJAIT)*, 13(1), World Scientific Publishing Company, Singapore, 2004, pp. 157-169.

[9] Salton, G., “Automatic text processing: the transformation, analysis and retrieval of information by computer”, Addison Wesley, Reading ,MA, 1989.

[10] H. Schmid, “Probabilistic Part-of-Speech Tagging using Decision Trees”, *Proceedings of International Conference on New Methods in Language Processing*, Manchester, 1994, pp. 44-49.

[11] Sinclair, J., “Corpus Concordance Collocation”. Oxford University Press, Oxford, 1991.

[12] R. Steinberger, “Cross-lingual Keyword Assignment”, *Proceedings of the XVII Conference of the Spanish Society for Natural Language Processing*, Jaén, 2001, pp. 273-280.

[13] L. Van der Plas, V. Pallotta, M. Rajman, H. Ghorbel, “Automatic Keyword Extraction from Spoken Text” *Proceeding of LREC 2004*, ELRA, Paris, pp. 2205-2208

[14] Text Encoding Initiative http://www.tei-c.org/

9 Fingerprints

9.1 State of the Art

Fingerprinting technologies are also called perceptual hashing technologies. “Perceptual hashing” indicates the common aspects with cryptographic hash functions while considering perceptual similarity. In the following subsections cryptographic hash functions are explained. Their drawbacks – when they are applied to multimedia content – are described. The advantages of perceptual hash functions are motivated.

An more detailed introduction that focuses on the common issues of perceptual hashing and cryptographic hashing can be found in DE4-5-1-Content-Protection-and-Supervision. Here, only the basic terms are shortly explained for providing a common understanding.

A hash functions maps a larger input space to a smaller one. The output of a hash function is called digest. It serves as a unique digital fingerprint for the input data. General prerequisites of cryptographic hash functions are:

- **Pre-image resistant**: given the hash h it should be hard to find the message m such that $h = hash(m)$ (one-way function).
- **Second pre-image resistant**: given an input m_1, it should be hard to find another input m_2 (not equal to m_1) such that $hash(m_1) = hash(m_2)$.
- **Collision-resistant**: it should be hard to find two different messages m_1 and m_2 such that $hash(m_1) = hash(m_2)$.
- **Random oracle property**: the hash function h behaves as a randomly chosen function.

Digital data streams are typically processed by computers. In contrast to these data streams multimedia data is perceived by humans. Obviously humans do not notice certain types of content modifications, e.g. the noise addition below the perceptual masking threshold. This property is exploited by lossy compression techniques, e.g. MP3 compression. Cryptographic hash functions however are bit-sensitive. Thus, using cryptographic hash functions for authenticating multimedia data is restricted to application where content is not processed or not available in different formats: Even format conversions are critical as the content modifications result in different hash values.

Perceptual hash functions

A perceptual hash function (fingerprinting function) should also satisfy the previously described requirements on cryptographic hash functions:

- Pre-image resistant
- Second pre-image
- Collision-resistant
- Random oracle property

Additionally, a perceptual hash function has to fulfil a different requirement which contradicts the prerequisites of the “second pre-image resistant” and the “collision-resistant”:

- **Perceptual similarity**: If two given inputs m_1 and m_2 are perceptually equal (similar) their corresponding hash values should be equal: $hash(m_1) = hash(m_2)$ (rsp. similar: $hash(m_1) \approx hash(m_2)$).

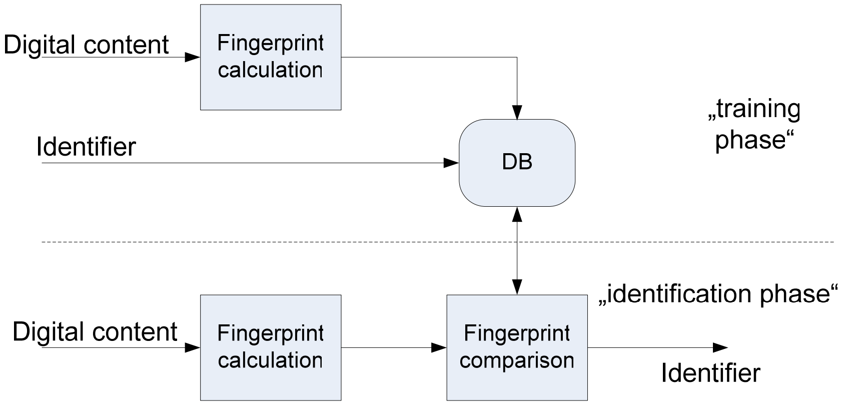
Perceptual similarity requires fingerprinting methods to be individually developed for different content types (like watermarking technologies).

General Requirements on fingerprinting technologies

- Discriminability
- Size
- Performance Characteristics
- Robustness
- Searching Complexity
- Security

Application: Content Description and Identification

To identify content, the unique descriptor calculated by fingerprinting and perceptual hashing techniques has to be stored in a database as shown in **Errore. L'origine riferimento non è stata trovata.**. This first phase is the training phase and can be compared with the training phase of a classifier. Having this image in mind, each piece of content is its own classification class. Content manipulations result in perturbations of the content and thus to potential "mis-classifications" in the identification phase.



Typical evaluation criteria for this application scenario are:

- Correct recognition rate
- False detection rate
- False alarm rate
- Feature extraction complexity
- Training complexity
- Retrieval complexity

9.2 Problems and Work Performed

The description of the problems and work done can be found in the following sections on the individual technologies. A description of the demonstration prototype together with implementation specific details is given in the Appendix A.

9.3 References

[Cano2002] P. Cano, E. Baltle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In IEEE Workshop on Multimedia Signal Processing, pages 169–173, 2002.

[Cox2001] I. J. Cox and J. paul M. G. Linnartz. Public watermarks and resistance to tampering, Aug. 30 2001.

[ECRYPT] ECRYPT, Network of Excellence. Recent collision attacks on hash functions: Ecrypt position paper. Technical report, ECRYPT, Nov 2004.

[Pre97] Bart Preneel. Cryptographic primitives for information authentication - state of the art. In *State of the Art in Applied Cryptography*, 1997.

[RFC2005] RFC-Editor. The request for comments (rfcs), 2005. http://www.rfc-editor.org/.

[Schn96] M. Schneider and S. F. Chang. A robust content based digital signature for image authentication. In *IEEE International Conference on Image Processing (ICIP'96)*, 1996.

[Schne96] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, second edition, 1996.

[WCHF2005] Wikipedia. Cryptographic Hash Function. http://en.wikipedia.org/wiki/Cryptographic hash function, June 2005.

10 Audio Fingerprints

10.1 State of the Art

Audio fingerprinting or content-based identification (CBID) technologies associate an audio signal to a much shorter mimetic sequence (the "fingerprint") and use this sequence to identify the audio signal (Cano, Gómez et al., 2002). Audio fingerprinting is best known for its ability to link unlabeled audio to corresponding metadata (e.g. artist and song name), regardless of the audio format.

As there is significant variance in audio data for perceptually similar content and direct comparison of the audio content itself is neither efficient not effective, a special hash method is seen as the best solution. The output of this hash function should be invariant under small perceptually insignificant modifications such that two audio excerpts are classified as "perceptually identical" if the human ear cannot distinguish between them.

10.1.1 Requirements

The prime demand of audio fingerprinting system is an efficient mechanism to establish the perceptual equality of two audio objects: not by comparing the (typically large) objects themselves, but by comparing the associated fingerprints (small by design).

In reality the design of an efficient mechanism depends heavily on the different applications. So during the design of an audio fingerprinting algorithm the various demands should be fulfilled, such as accuracy, reliability, robustness, security, versatility, scalability, and complexity. Generally, the fingerprint should be (Cano, Batlle et al., 2002):

- A perceptual digest of the recording. The fingerprint must keep as much acoustically relevant information as possible and should allow discrimination between large numbers of fingerprints.
- Invariant to distortions. The fingerprint must be accurately connected with its original recording, regardless of the level of compression and distortion or interference in the transmission channel.
- Compact. Because a large number (millions) of fingerprints need to be stored and compared, a small-sized representation is necessary. At the same time an excessively short representation might not be sufficient to discriminate among recordings, affecting thus accuracy, reliability and robustness.
- Easily computable. In order to reduce complexity, the extraction of the fingerprint should be easy to compute.

According to those demands 5 basic parameters must be focused on (Haitsma et al., 2002):

- Robustness: An audio clip must be identifiable even after severe signal degradation. The fingerprint should use perceptual features that are invariant in order to achieve this. Ideally, severely degraded audio still leads to very similar fingerprints. The False Rejection Rate (FRR), i.e. the probability that two signals are "equal", but not identified as such, can be used as robustness metrics.
- Reliability: Results must be consistent for variety of input signals. The False Acceptance Rate (FAR), i.e. the rate at which audio signals are incorrectly considered "equal", is always used to specify reliability.
- Fingerprint size: Fingerprints are usually stored in RAM memory to enable fast searching. Thus the fingerprint size should be kept to a minimum.

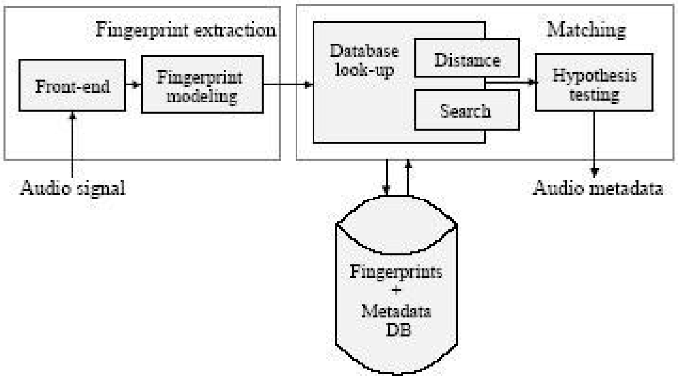
- Granularity: The length of an audio excerpt to identify an audio clip is an application dependent parameter.
- Search speed and scalability: The time needed to find a fingerprint in the fingerprint database is a crucial parameter for commercial deployment of audio fingerprint systems.

These five basic parameters are all intertwined. For instance, search speed will increase when a more robust, smaller fingerprint is designed. On the other hand the smaller fingerprint may impact the reliability parameter.

10.1.2 Basic Schema

Although the design of an audio fingerprinting system depends on the application scenario, there are two components that are common all audio fingerprinting systems: the fingerprint extraction and the matching algorithm (Cano et al., 2002).

Errore. L'origine riferimento non è stata trovata. shows a content-based audio identification framework as described by Cano et al., 2002. The first block is the processing stage, which actually computes a fingerprint, while the second block involves sophisticated search algorithms to scan a database of previously computed fingerprints for matches. In order to fulfil the requirements of audio fingerprinting, the extraction step consists of a front-end and a fingerprint modelling block, which implies a trade-off between dimensionality reduction and information loss. The fingerprint model block is supported by the front-end, which converts an audio signal into a sequence of relevant features, while the fingerprint model block defines the final fingerprint representation, e.g.: a vector, a trace of vectors, or a codebook. Distance measures and searching algorithms are utilized during the database look-up step. Finally, the hypothesis testing component gives us a reliability measure that indicates the quality of the match.



In the following sections we will describe the modules of the basic schema in more detail.

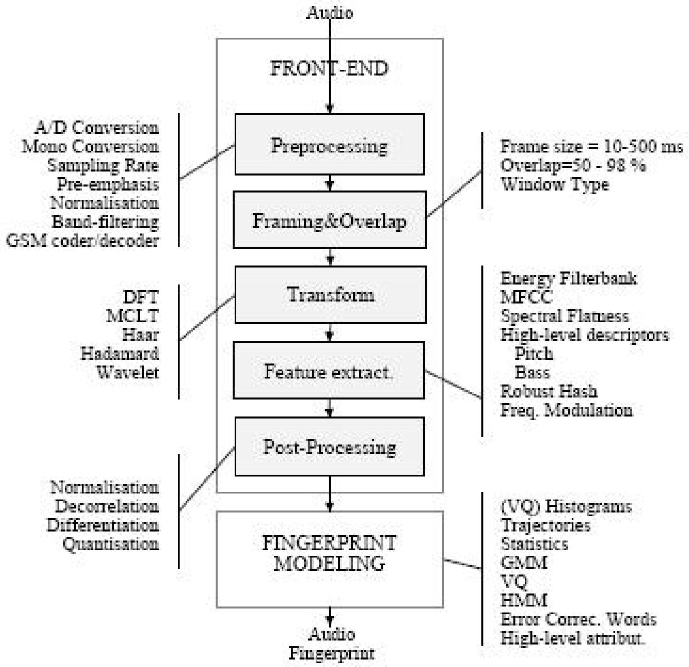
Front-end

The front-end consists of 5 sub-modules, which are shown in **Errore. L'origine riferimento non è stata trovata.**6:

- Pre-processing: The audio stream is digitized (if necessary) and will be subject to operations such as re-sampling, normalisation, and band-filtering, resulting in a canonical presentation of the data.
- Framing and Overlap: The signal is divided into frames such that the signal within the frame can be regarded as stationary over the frame interval. The frames have some time overlap to assure robustness to time shifting.
- Linear Transforms: In order to significantly reduce the redundancy within the original signal and to facilitate further processing steps, linear transforms are applied to convert the original data to a new domain. Standard transforms from time to frequency domain can ease efficient compression, noise

removal and subsequent processing. The Fast Fourier Transform (FFT) is one of the most popular transformations in this scenario.

- Feature Extraction: Usually based on the time-frequency representation mentioned above (framing and FFT), a variety of algorithms can be applied to reduce the dimensionality of the data and, at the same time, to increase the invariance to distortions. The outcome of this processing stage is the final feature vectors.
- Post-processing: The most common post-processing step is to apply low resolution quantization to the features: ternary or binary, which can ease hardware implementations and make a convenience of subsequent parts of the system.



Fingerprint Models

The objective of this phase is to further reduce the fingerprint size. Because the fingerprint modelling block always consists of a sequence of feature vectors, which are calculated based on a frame, it is suggested to exploit redundancies in the frame time vicinity, inside a recording and across the whole database. Fingerprint models are the preconditions to select the distance metric and design indexing algorithms for fast retrieval. While a very concise form of fingerprint is achieved by summarizing the multidimensional vector sequences of a complete audio stream (or a fragment of it) into a single vector, fingerprints can also be formed through the concatenation of sequences (traces, trajectories) of features.

Distance Metrics, Search Methods and Hypothesis Testing

According to a predefined distance metrics both the set of reference patterns (the fingerprints stored in the database) and the test pattern (the fingerprint extracted from the data stream to be classified) are compared. Distance metrics are dependent on the type of fingerprint model. For instance, a correlation is used to compare vector sequences, while the hamming distance is used to compare single vectors.
The basic goal is to build a data structure which reduces the number of distance computations required for a query. A good search method should be fast, correct, memory efficient and easily updateable.
Hypothesis Testing: This last step decides, based on a threshold value, if a fingerprint is present in the databases.

10.1.3 Existing Algorithms

Different features have been proposed and implemented for the calculation of audio fingerprints. The can be categorised in two groups:

- Signal characterising features are derived from the signal itself. They are low-level features, which are no directly perceived by human listeners. However, they change if the audio signal changes. Examples for these features are loudness, energy, sharpness, spectral centroid, or zero crossing rate.

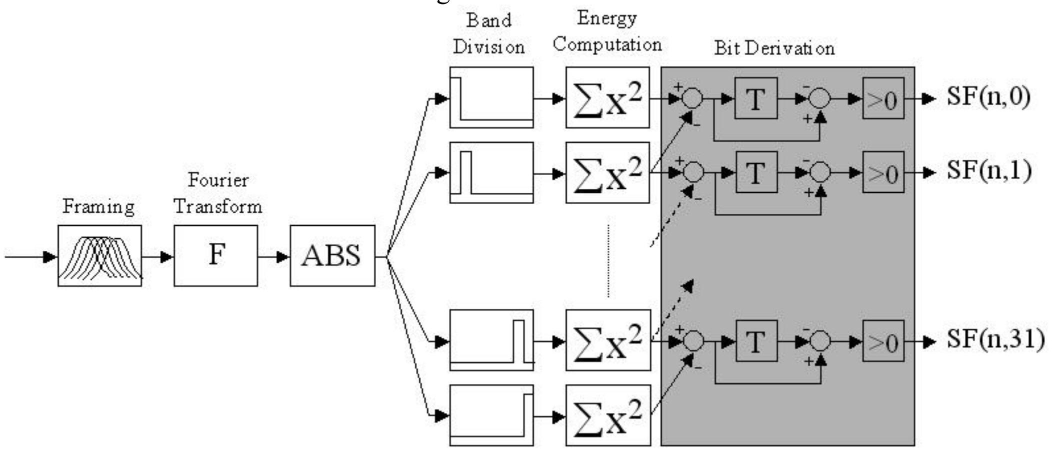
- Perceptually relevant features are related to the sound perceived by humans. Like tone-likeness or tonality. These features, which are perceptually meaningful, are supposed to be more robust with respect to distortions.

Signal Characterising Features

A good example for an audio fingerprinting algorithm based on (statistical) signal characterising features is the algorithm presented in [Haitsma2002a]. The following steps can be identified:

1. The audio stream is segmented into overlapping frames (time segments).
2. For each frame:
   a. The coefficients are transformed in the Fourier-domain and the absolute values of the Fourier coefficients are calculated.
   b. The Fourier coefficients are separated into different bandwidths (frequency segments). For each segment the energy is computed.
   c. Instead of directly considering the energy values, the energy differences of neighbouring frequency bands are calculated and their temporal change (increasing/decreasing) is chosen for a single fingerprint value.

The overall architecture is shown in the next figure.

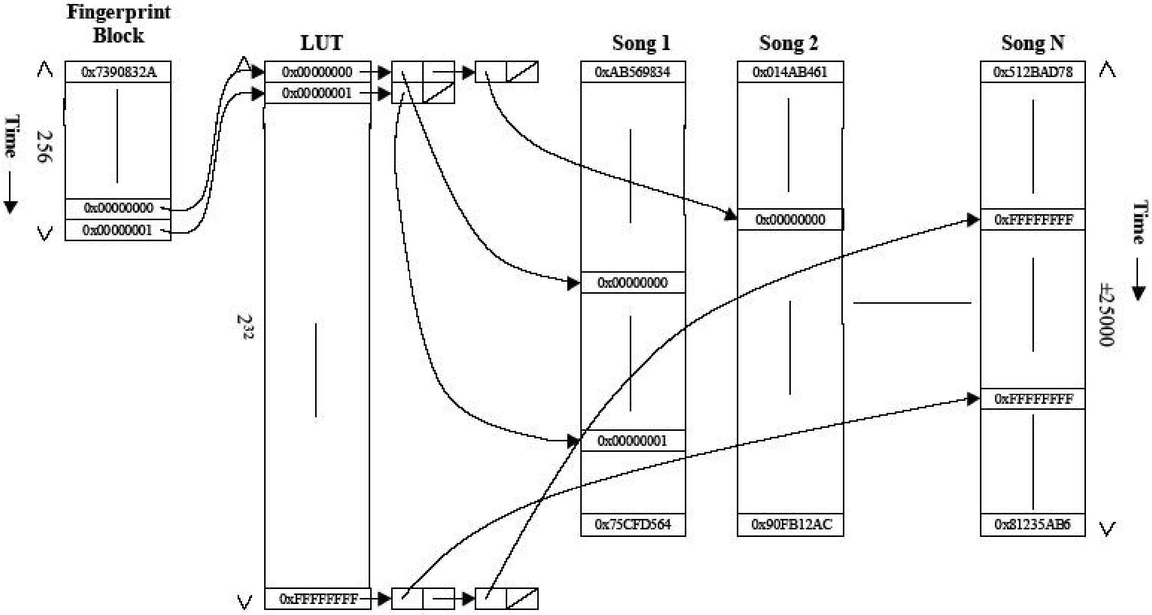


The resulting hash values for each window are called a sub-fingerprint. The retrieval itself is based on a "collection" of sub-fingerprints, which are called the "fingerprint".

The retrieval of the algorithm based on a complex database structure. The complexity of the database structure is derived from the requirement of breaking "the curse of dimensionality". Only by limiting the high-dimensional search space (time-) efficient retrieval can be guaranteed. The retrieval process consists of the following steps.

1. Each (sub-) fingerprint is used as an index to a hash table. By this step the possible song candidates are pre-selected. Also the temporal position of fingerprint in the song is identified.
2. This pre-selection together with the position identification strongly limits the search space. Based on this, the input fingerprint can be compared with the stored fingerprints.

This process together with the corresponding database structure is shown below.



Perceptually Relevant Features

As described in [Herre2001] perceptually relevant features, which indicate tone-likeness or tonality include:

- Spectral peaks indicate the presence of a tonal component in the Power Spectrum Density (PSD).
- A high predictability of the spectral coefficients of the audio signal indicate tonality.
- The Spectral Flatness Measure (SFM) quantifies the tonal quality, which is defined as the ratio of the geometric mean to the arithmetic mean of the power spectral density components:

$$SFM(s(t)) = \frac{\left[\prod_{k=0}^{N-1} S_{xx}(f)\right]^{\frac{1}{N}}}{\frac{1}{N}\sum_{k=0}^{N-1} S_{xx}(f)}$$

- The Spectral Crest Factor (SCF) is the ratio of the largest PSD coefficient and the mean PSD value in a frequency band. The Crest Factor "is equal to the peak amplitude of a waveform divided by the RMS value. The purpose of the crest factor calculation is to give an analyst a quick idea of how much impacting is occurring in a waveform." [Friedman05]. Thus, the SCF

According to [Herre2001] the most promising feature candidates are the SFM and the SCF. Alternative variations of these features are also suggested in [Herre2001], which are called a "family of flatness oriented spectral features".

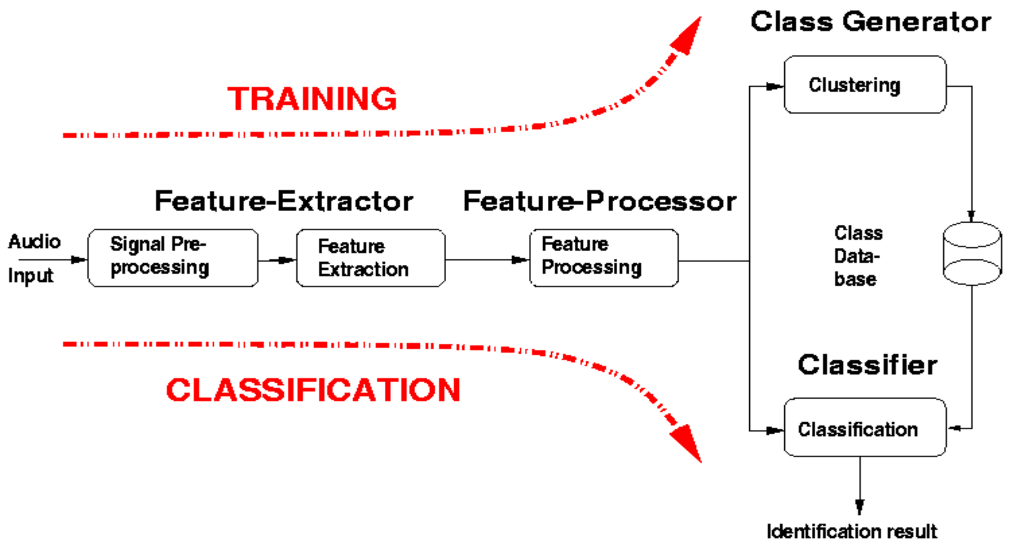
In [Herre2002] the SFM and the relation to the MPEG-7 standard is described:

- The "AudioSpectrumFlatness Low Level Descriptor (LLD)" defines how to extract this feature from the audio signal.
- The "AudioSignature Description Scheme (DS)" defines how an interoperable hierarchy of scalable fingerprints can be established.

The AudioSignature DS considers the following three parameters:

- The so-called temporal scope identifies the start position and the length where the feature has to be or was extracted.
- The temporal resolution is a kind of grouping parameter for single frames of 30ms. The SFM are computed and statistically summarized for one group.
- The spectral coverage or spectral bandwidth of the identifies the frequency bands that have to be selected for the feature calculation.

The retrieval process described in [Herre2001, Herre2002] groups several features – these groups are called super vectors (c.f. fingerprints and sub-fingerprints in the previous description) – and use statistical measures for the retrieval of the nearest neighbour. No details are given in [Herre2001, Herre2002]. However, it seems that techniques from pattern recognition and string matching are applied in the "classification" or "retrieval phase" to identify the "best approximation" stored in the database during the "training phase". The different steps involved are shown in the figure below.



Further developments

Besides the above described algorithms different developments and improvements exists.

In [Bardeli04] an algorithm is proposed that is similar to [Herre2001] or [Haitsma2002]: The audio data is segmented. For each segment a binary string is calculated. Retrieval is based on a clustering algorithm. The segmentation is based on prominent local maxima from the spectrum. For the calculation of the binary sequence the equally spaced frequency bands are chosen and the energies in these sub-bands are considered.

Burges et al. [Burges02, Burges03] derive features for the identification of audio by using an oriented PCA as suggested in [Diama96]. This method for Distortion Discriminant Analysis (DDA) is iteratively applied three times to the input frames of the audio stream. In the "learning phase" a transformation of the input data is identified that maximizes inter-class variance and minimizes intra-class variance. In the "classification phase" the identified transformation(s) is (are) applied to the input data. The distance to stored dimensionality reduced features are used for the audio identification process.

The Mel-Cepstrum Coefficients (MFCC) are proposed in [Batlle02] for the calculation of the features. A Hidden Markov Model (HMM) is used for the calculation of the feature vector.

The audio fingerprinting method suggested by [Lu02] is based on wavelet coefficients: As described in [Mallat92] finding the local wavelet maxima is equivalent to multi-scale Canny edge detection. Thus, an approximately reconstruction of a signal is possible from its local wavelet maxima.

The problem of the linear speed change is the focus of the research in [Seo02]. The authors propose to use the invariance properties of the Fourier-Mellin Transform on the method presented in [Haitsma2002, Haitsma2002b].

10.2 Problems

It can be clearly seen that the focus of existing audio fingerprinting systems is on content identification. The application of content authentication/content verification has not been considered so far.

A detailed benchmarking of the fingerprinting system regarding verification is missing. The decision threshold during the verification step has to be evaluated regarding the acceptable distortion for different kind of manipulations in the envisaged application scenario. Regarding the robustness of the audio fingerprint system against time scale modification further research is necessary.

10.3 Work Done

In general every audio fingerprinting system consists of an feature extraction and matching algorithm. The following sections describe the various building blocks of the audio fingerprint system developed by IGD.

Feature Extracting Block

Data preparation

The audio data is converted to a raw mono signal with 16 bit amplitude resolution and a sampling rate of 44.1 kHz. Pre-processing of the audio samples like normalization of the amplitudes can be optionally applied but is currently not necessary. The input architecture of the audio fingerprint system will uses parts of the psychoacoustic model of MPEG 1 Layer I [Pan95].

Segmentation into Frames

The basic algorithm splits the audio track into time slices (frames) for extracting parts of the whole fingerprint, the so-called sub-fingerprints. The number of samples per frame is 512. The number of samples is determined by the performance of linear transform (see below) applied to the frame and the usage of the underlying psychoacoustic model (MPEG 1 Layer I). In order to minimize the leakage effect due to the edges of the frames the input block is multiplied with a weighting window. The Hanning window is used as a weighting function.

Overlapping of the Frames

Due to the fact that a misalignment of the frames during feature extraction and matching can occur, the individual frames are overlapped by 384 samples to introduce a correlation along the time axis. The overlapping transforms the correlation of the audio tracks within a few milliseconds to correlation within the fingerprints, which can be utilized to improve the robustness against shifting during the matching of the fingerprints of two different audio tracks.

Transformation of the Individual Frames

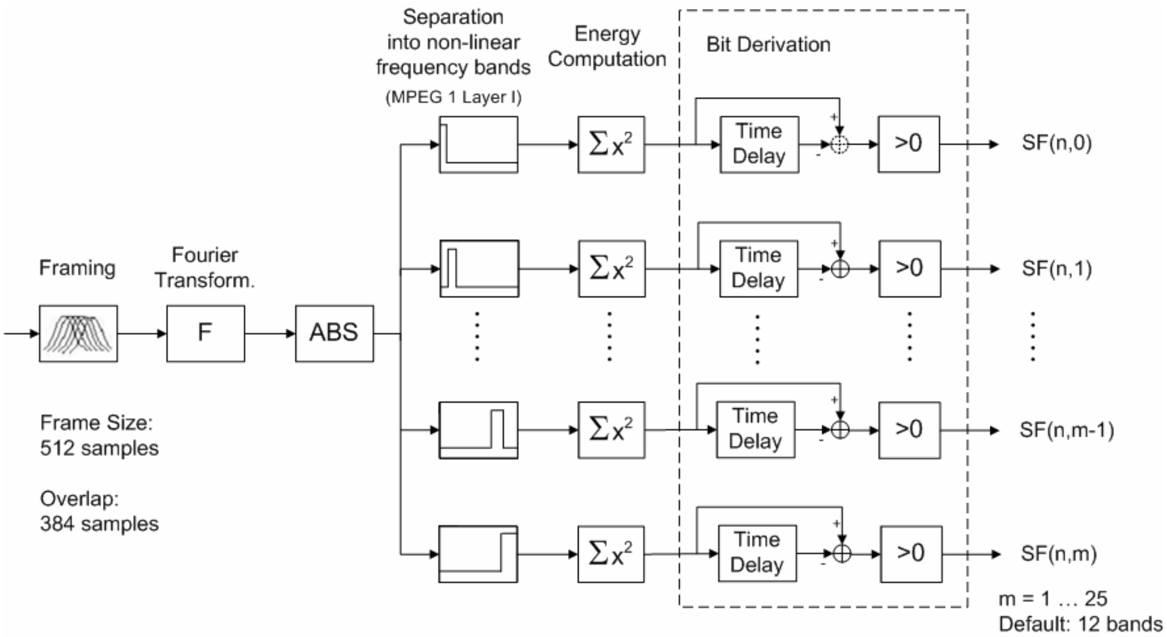
Due to the fact that most of the signal processing like lossy compression, filtering, adding noise, etc. can be described in the fourier domain and the fact that very fast implementations exist, it seems to be the most appropiate working domain. Furthermore perceptual relevant characteristics representing suitable features are often computed in the Fourier domain. The successive frames in the time domain are transformed via a Fourier transformation in order to compute a time-frequency representation of the audio signal.

Extraction of the Features

The feature extraction procedure reduces the dimensionality and increases the robustness against distortions. Furthermore knowledge about the human auditory system in order to extract perceptual relevant parameters are exploited in this step. Therefore a mapping is performed between the Fourier domain and the Bark scale. The spectrum is analyzed in critical bands, which are investigated regarding their energy. This includes the calculation of the energy in the frequency domain and the difference between successive frames along the time axis.

Processing and Modelling of the Features

The extraction of the energy differences is performed for every pair of successive audio frames as shown in the next figure. The individual sub-fingerprints are concatenated to construct the fingerprint of the whole audio track. In order to calculate a more compact representation of the audio fingerprint and to provide a higher robustness against distortions the calculated energy components are quantized to a binary representation.



Matching Algorithm and Distance Metric

The resulting audio fingerprint is modelled by a combined binary vector being composed of binary vectors for the individual sub-fingerprints. The fingerprint template used determines the distance metric used during the comparison of the fingerprints from two audio tracks.

Matching Algorithm

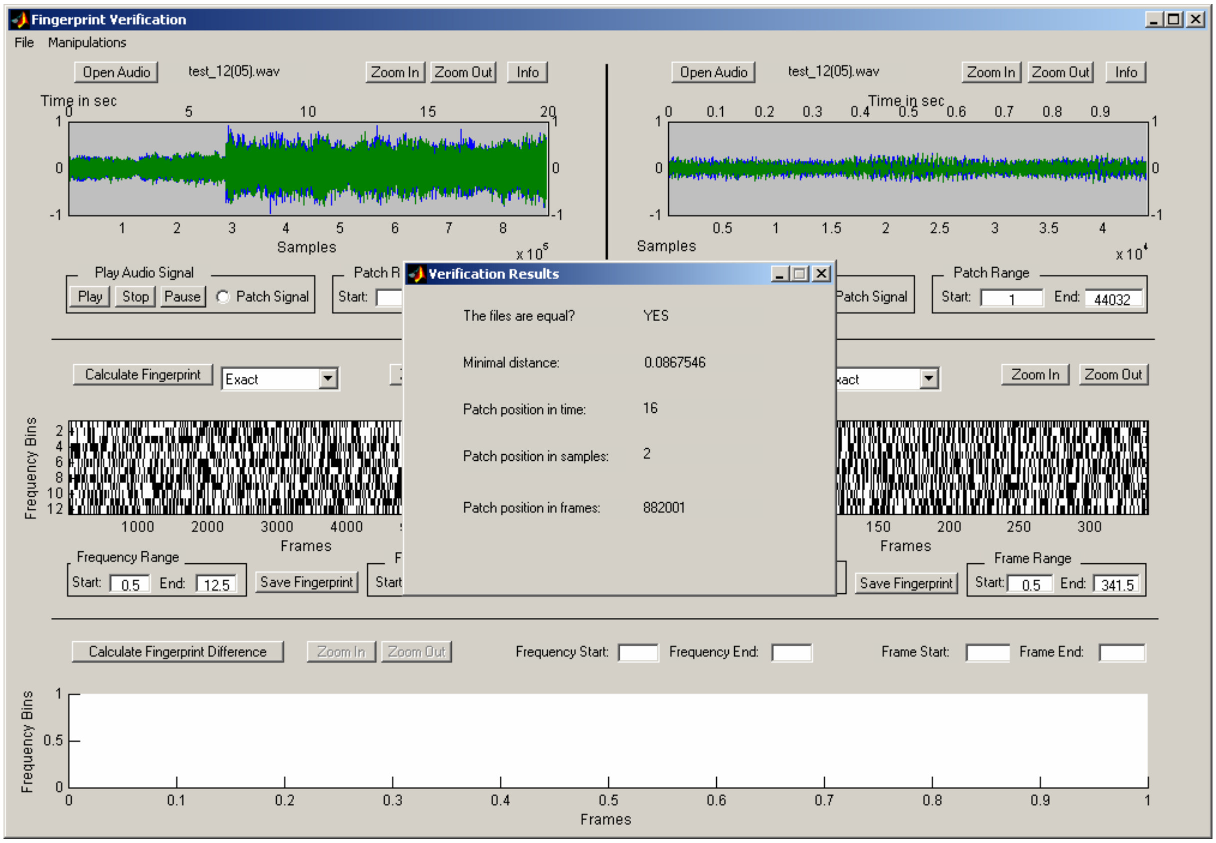
One of the fundamental requirements of audio fingerprint systems is the ability to accurately identify the audio tracks from excerpts of a few seconds. This requirement of robustness against cropping corresponds to the minimum watermark segment and is often denoted as granularity. The fingerprints for the individual frames can be used as indexing mechanism in order to identify the position in the song. The binary vectors are compared by calculating the normalized Hamming distance. This distance metric is equivalent to the BER if the fingerprints of the same audio track are compared.

Decision Block

The decision block uses the distance metric in order to formulate the hypothesis whether the audio track is the same as the presented one or not. The decision threshold will depend on the false acceptance and false rejection rate acceptable for the application and the discrimination ability of the fingerprint system which is directly related to the number of audio tracks in the system. The decision threshold for different kinds of manipulations is currently evaluated.

Demonstration Graphical User Interface

The implemented prototype functionality was integrated in a Graphical User Interface (GUI). This GUI allows an simple demonstration of the capabilities of the implemented algorithm.



Fingerprint Verification

test_12(05).wav

test_12(05).wav

Exact

Frequency Range Start: 0.5 End: 12.5

Patch Range Start: 1 End: 44032

Frame Range Start: 0.5 End: 341.5

Verification Results

The files are equal? YES

Minimal distance: 0.0867546

Patch position in time: 16

Patch position in samples: 2

Patch position in frames: 882001

10.4 References

[Batlle02] Batlle, E.; Masip, J. & Guaus, E. Automatic song identification in noisy broadcast audio 2002

[Bardeli04] Bardeli, R. & Kurth, F. Robust Identification of Time-Scaled Audio Proceedings of the AES 25th International Conference on Metadata for Audio, 2004

[Burges02] Burges, C.; Platt, J. & Jana, S. Extracting noise robust features from audio data IEEE International Conference on Acoustics, Speech, and Signal Process, 2002, 1021-1024

[Burges03] Burges, C.; Platt, J. & Jana, S. Distortion discriminant analysis for audio fingerprinting IEEE Transactions on Speech and Audio Processing, 2003, 11, 165-174

[Cano02a] Cano, P. Batlle, E., Kalker, T. and Haitsma, J (2002), A Review of Algorithms for Audio Fingerprinting. In *International Workshop on Multimedia Signal Processing*, *2002.*

[Cano02b] Cano, P. Gómez, E. Batlle, E. Gomes, L. Bonnet, M., (2002)., Audio Fingerprinting: Concepts and Applications. *Proceedings of 2002 International Conference on Fuzzy Systems Knowledge Discovery*

[Diama96] Diamantaras, K.I. & Kung, S.Y. Principal component neural networks: theory and applications John Wiley & Sons, Inc., 1996

[Friedman05] Friedman, A.: What is the Crest Factor, http://www.dliengineering.com/downloads/crest%20factor.pdf

[Herre2001] Herre, J.; Allamanche, E. & Hellmuth, O. Robust matching of audio signals using spectral flatness features, IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics, 2001, 127 – 130

[Herre2002] Herre, J.; Hellmuth, O. & Cremer, M. Scalable robust audio fingerprinting using MPEG-7 content description IEEE Workshop on Multimedia Signal Processing, 2002, 165 - 168

[Haitsma2002] Haitsma, J., and Kalker, T. (2002), A highly robust audio fingerprinting system. *International Symposium on Musical Information Retrieval, 2002*

[Haitsma2002b] Haitsma, J.; Kalker, T. & Oostveen, J. An efficient database search strategy for audio fingerprinting IEEE Workshop on Multimedia Signal Processing, 2002

[Lu02] Lu, C. Audio fingerprinting based on analyzing time-frequency localization of signals, IEEE Workshop on Multimedia Signal Processing, 2002

[Mallat92] Mallat, S. & Zhong, S. Characterization of Signals from Multiscale Edges IEEE Trans. Pattern Anal. Mach. Intell., IEEE Computer Society, 1992, 14, 710-732

[Seo02] Seo, J.S.; Haitsma, J. & Kalker, T. Linear speed-change resilient audio fingerprinting Model based Processing and Coding of Audio, 2002

11 Image Fingerprints

11.1 State Of The Art

Schneider and Chang proposed the first image fingerprinting methods [Schn96]. This method is based on image histograms. Its development aim was the authentication of digital content. Different methods have been proposed since then.

Methods that have been recently published try to increase the robustness of the calculated fingerprints by pre-processing the content before fingerprint calculations. These transformation can range from wavelet transforms, which simulate the human perception, to other complex mathematical transformation to increase the robustness of the calculated fingerprints against rotation, scaling and translation (R,S,T).

The so-called “soft hash algorithm” as proposed by Lefebvre and Macq [Lefe03] is based on the Radon transform and on Principal Component Analysis (PCA). The medium point of each projection is used as a fingerprint. An improved version was presented by Seo [Seo03].

A method that is based on image statistics was presented by Datta et al. [Datta03]. These pixel statistics are simple moments including area, extent, and eccentricity. They are calculated for image segments. Thus a region segmentation of the input image has to be calculated prior to the fingerprint calculation.

Lu et al. [Lu04] proposed a method inspired from the idea of Bas' method [Bas02], which is based on a feature-driven triangulation and which resistant to almost all geometric distortions introduced by Stirmark.

11.2 Problems

Methods based on image intensities like histogram based methods generally give good results. They depend on the intensity of the image. Thus they are sensitive against colour changes like grayscale reduction. Also, the effect of gaussian blur and lossy compression cannot be neglected practically.

Methods based on radon transform are based on search and energy modification for robustness against rotation and scaling. Also, the published methods don't provide keyed hash function, which is important for content authentication. In addition to that, the improved method by [Seo03] as well as in the original method [Lefe03] lack in the robustness of the fingerprint against geometrical modifications strongly depend on the limited invariance property of the Radon transform.

Methods based on image segmentation are influenced by operations affecting the segmentation. Thus, one can expect, that colour reduction as well as low- or high-pass filter operations affect the segmentation. In [Datta03] experimental results are not specified. This makes a comparison and evaluation difficult.

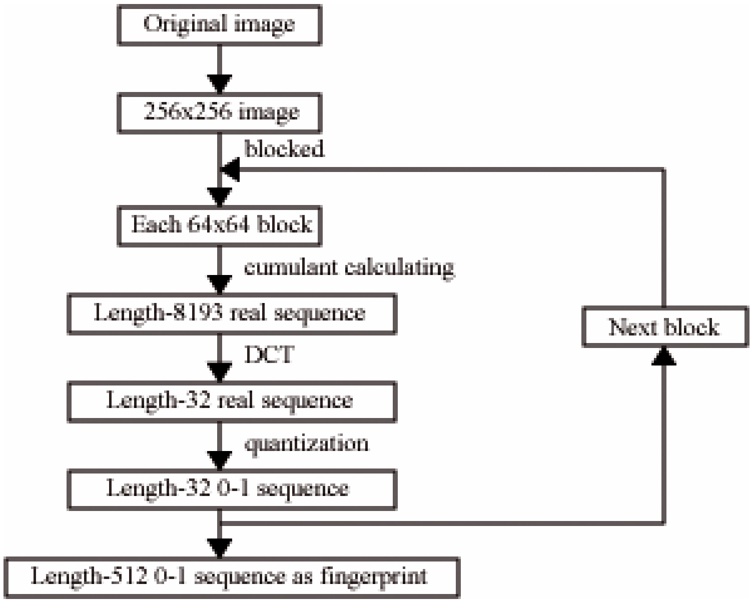
Methods based on image triangulation are influenced by operations affecting the triangulation. Geometrical attacks - especially cropping - cannot not well settled due to the destruction of mesh triangulation. In addition, problems can be foreseen with small sized images as robust mesh extraction is difficult.

11.3 Work Done

In general every audio fingerprinting system consists of an feature extraction and matching algorithm. The following sections describe the various building blocks of the audio fingerprint system developed by IGD.

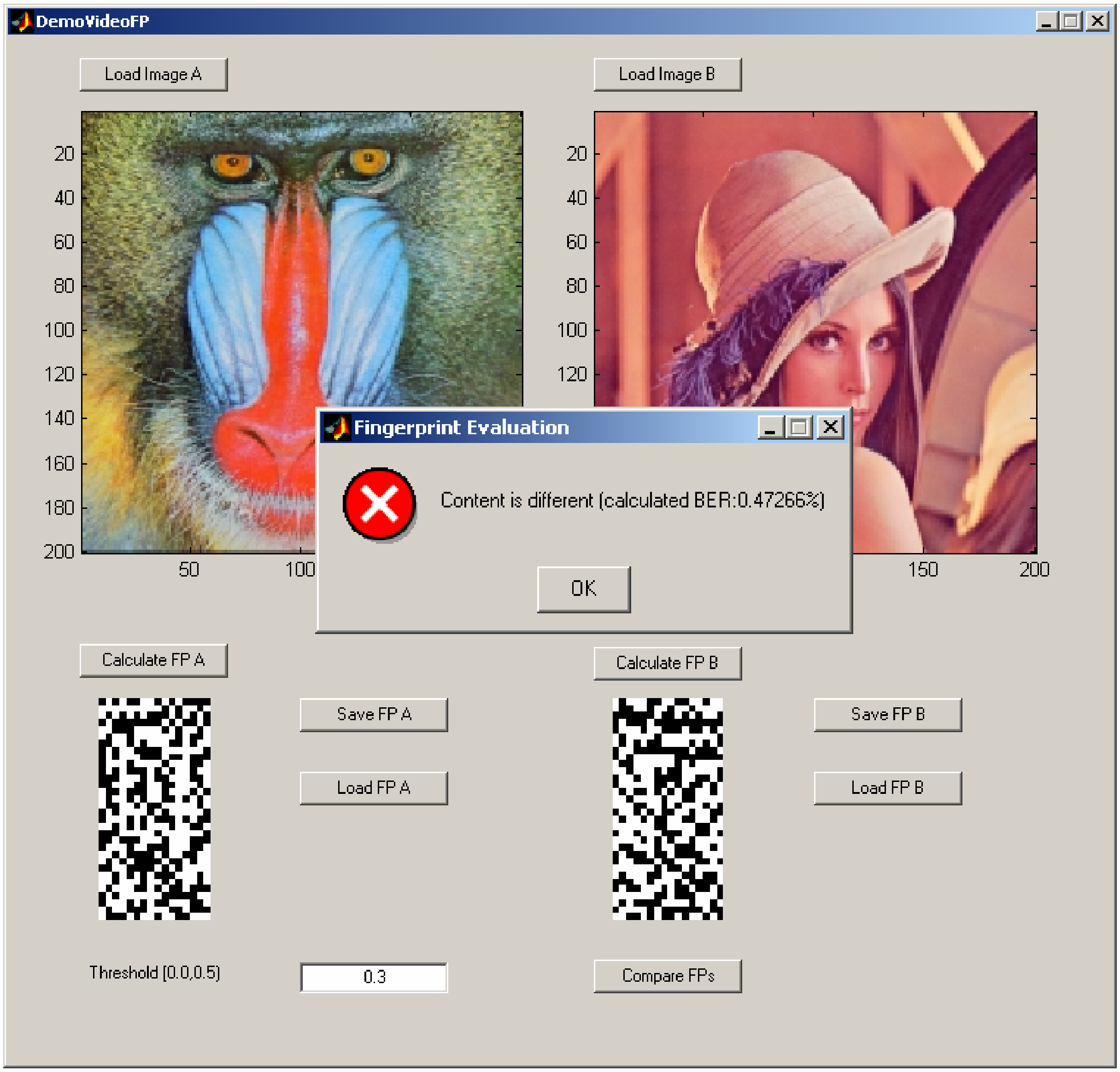
1. Identification of suitable features for the calculation of image fingerprints: The basic features are higher order statistics/moments: cumulants. Cumulants are often used in modern signal analysis. In some signal processing applications, moments provide some inherent advantage, especially when objects are assumed as Gaussian signals. However, many real world signals are not Gaussian signals and techniques used in traditional spectrum analysis are not longer adaptable. In these cases cumulants are introduced as a measure to evaluate the non-Gaussianity of a signal. Since cumulants are very stable even in noise-dominating environments, they are often taken as a method to extract the signal from noise.

2. Development and implementation of an image fingerprinting algorithm based on cumulants. The scheme of the developed algorithm is shown in the below. The original image is rescaled and tiled. For each tile the forth order statistics (cumulants) are calculated. These cumulants “summarize” the dependencies of four pixels in the individual tile/block. As cumulants are real values, the resulting real valued vector is binarized resulting in a 512 bit fingerprint that is characteristic for each image.



3. First analysis of the implemented algorithm: Based on the some internal test images the developed algorithm was analysed and its performance has been successfully compared with other state-of-the-art algorithms.

4. The implemented prototype functionality was integrated in a Graphical User Interface (GUI). This GUI allows an simple demonstration of the capabilities of the implemented algorithm.



Content is different (calculated BER:0.47266%)

0.3

11.4 References

[Schn96] M. Schneider and S. F. Chang, “A Robust Content Based Digital Signature for Image Authentication” , ICIP'96, IEEE, 1996

[Lefe03] F. Lefebvre, J. Czyz and B. Macq, “A Robust Soft Hash algorithm for Digital Image Signature", ICIP'03, IEEE, 2003

[Seo03] J.S., Seo, J. Haitsma, T. Kalker and C. D. Yoo, "Affine transform resilient image fingerprinting", ICASSP'03, IEEE, 2003

[Datta03] A. Datta, N. D. V. Lobo and J. Leeson, "Novel Feature Vector for Image Authentication", International Conference on Multimedia and Expo (ICME), IEEE, 2003

[Lu04] C. S. Lu, C. Y. Hsu, S. W. Sun and P. C. Chang, "Robust Mesh-based Hashing for Copy

Detection and Tracing of Images", International Conference on Multimedia and Expo (ICME): special session on Media Identification, IEEE, 2004

[Bas02] P. Bas, J. M. Chassery and B. Macq, "Geometrically invariant watermarking using feature points", IEEE Trans. on Image Processing, Vol. 11, 2002

12 Video Fingerprints

12.1 State of the Art

As the main characteristic of video is time-dependency, the general video fingerprinting architecture has much in common with the general audio fingerprinting architecture. The difference lies in the feature derived from the visual content. Two different approaches can be identified:

- Methods based on features which are based on very low-level features
- Methods based on features which have a

As described in [Caspi2004] only few methods consider the temporal structure of a video. Instead the extract key-frames form the video. For these key-frames the fingerprint is calculated.

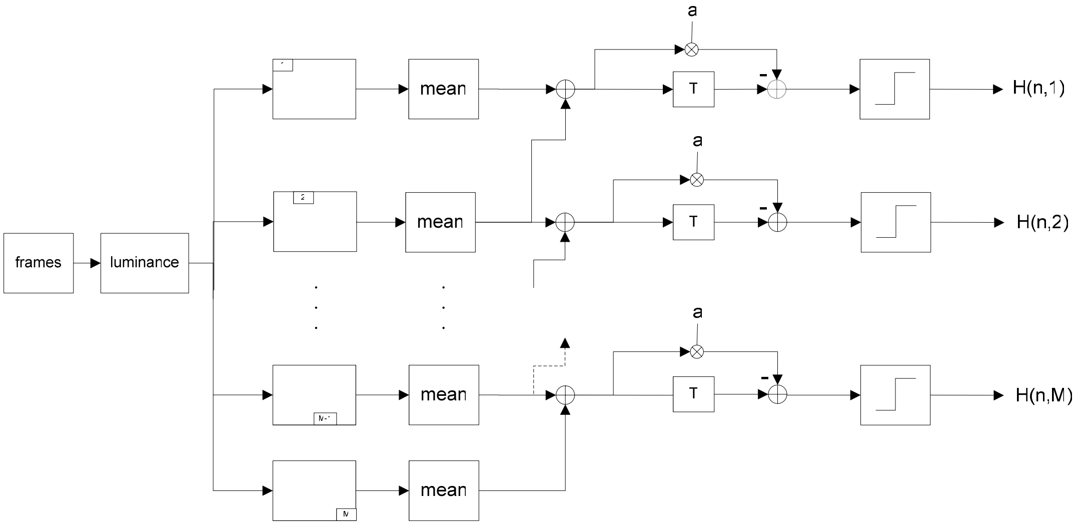
12.1.1 Video Fingerprints based on Frame Statistics

The method described in [Oost02] extracts the fingerprint in the spatial-temporal domain due to performance reasons: DCT and DFT are considered as complex and therefore costly operations. Thus, the algorithm is based on simple statistics, like mean and variance. These statistics are calculated in image regions.[2]

The algorithm is shown in **Errore. L'origine riferimento non è stata trovata.**. Different individual steps can be identified:

- For each frame the luminance is chosen for the calculation of the fingerprint. (This algorithm is not necessary restricted to the luminance channel).
- Each frame is divided into spatial blocks.
- For each spatial block the mean luminance is calculated.
- The spatial difference with a spatial neighbouring block is calculated.
- The temporal difference with the previous calculated block difference is calculated.
- The temporal difference is binarized and thus directly contributes to as a single bit to the fingerprinting vector.

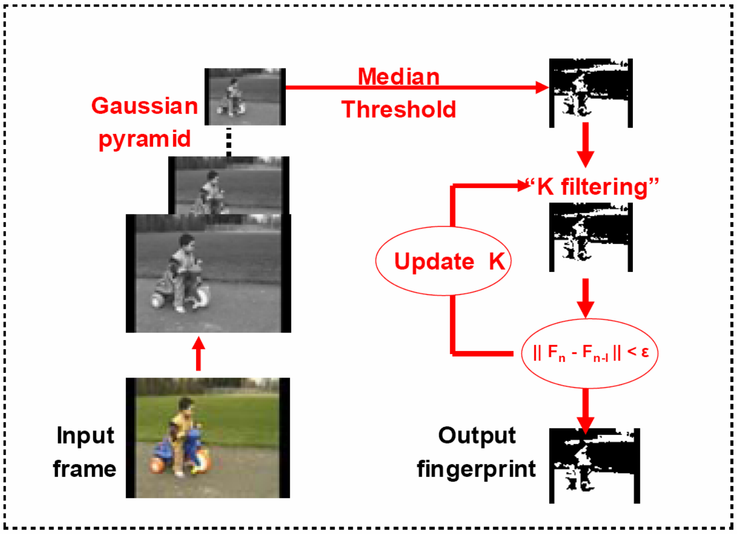
[2] In [Oost02] the authors explicitly state that the focus of this algorithm is on robustness and not on security.



For retrieval the same authors proposed an efficient data structure to speed up searching in databases. This data structure considers the temporal dependencies: A lookup-table (LUT) allows to identify the songs which potentially include the calculated fingerprint. A hypothesis test identifies or rejects the candidates by considering the temporal dependency of the video: For each song the consecutive fingerprints are stored.

Summarized, the previously described method considers spatio-temporal block differences. Other methods are proposed which are also based on the tiling of individual frames. The difference lies in the prposed features. For example, the method proposed in [Muce04] is based on simple statistics. It uses the characteristic variance maxima or minima positions in block.

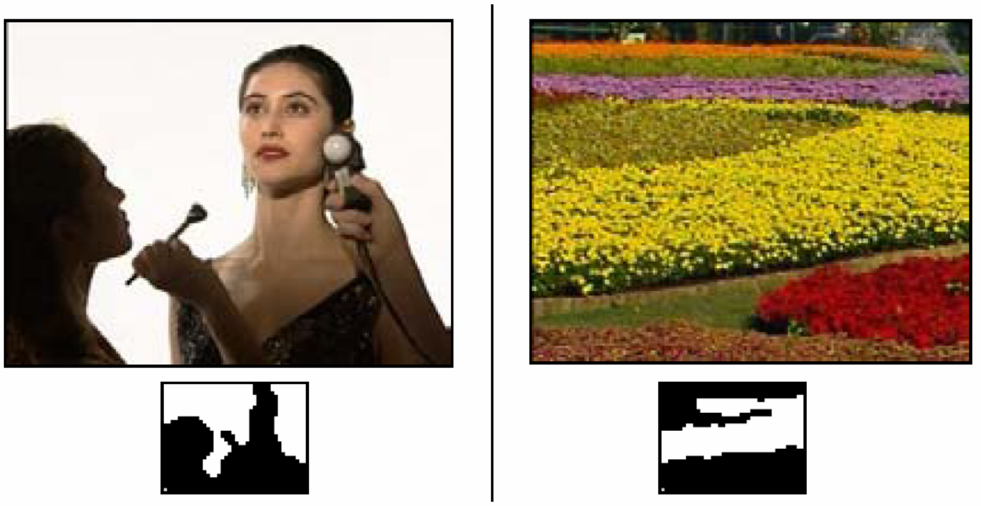
12.1.2 Video Fingerprints based on (Key) Frame Content



In contrast to the previously described methods, which consider temporal changes, there is a second class of identification algorithms, which are based on the content of key-frames.

For example, the algorithm proposed in [Caspi2004] is based on the intensities of key-frame. Key frames can be typical shot frames and partially extracted automatically [Hanj2002]. On this input frame a Gaussian pyramid is calculated (multi-scale representation). The low resolution is thresholded (binarised). To increase the robust of this method a so-called "k-filtering" based on morphological operations is iteratively applied until a equilibrium is reached.

The result of this process is a binary (two-dimensional) vector. This vector is the fingerprint (perceptual hash) corresponding to the key frame. An example of different resulting bit-representations is shown below.



In [Yang04] a so-called "locality sensitive hashing" (LSF) is proposed. The LSF hashing is based on approximate ε-nearest neighbour search. This solves the problem of similarity search.

12.2 Problems

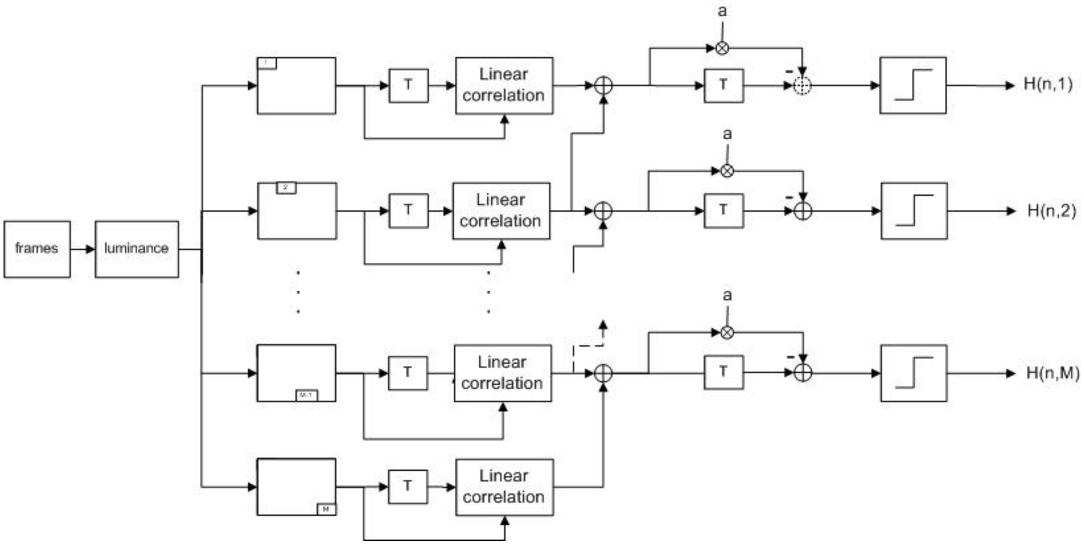
As for audio fingerprinting, the focus of existing video fingerprinting systems is on content identification. The application of content authentication/content verification has not been considered so far. Also, a detailed benchmarking of the fingerprinting system regarding verification is missing. The decision threshold during the verification step has to be evaluated regarding the acceptable distortion for different kind of video processing operations in the envisaged application scenario.

12.3 Work Done

In general every video fingerprinting system consists of a feature extraction and matching algorithm. The following sections describe the various building blocks of the video fingerprint system developed by FHGIGD.

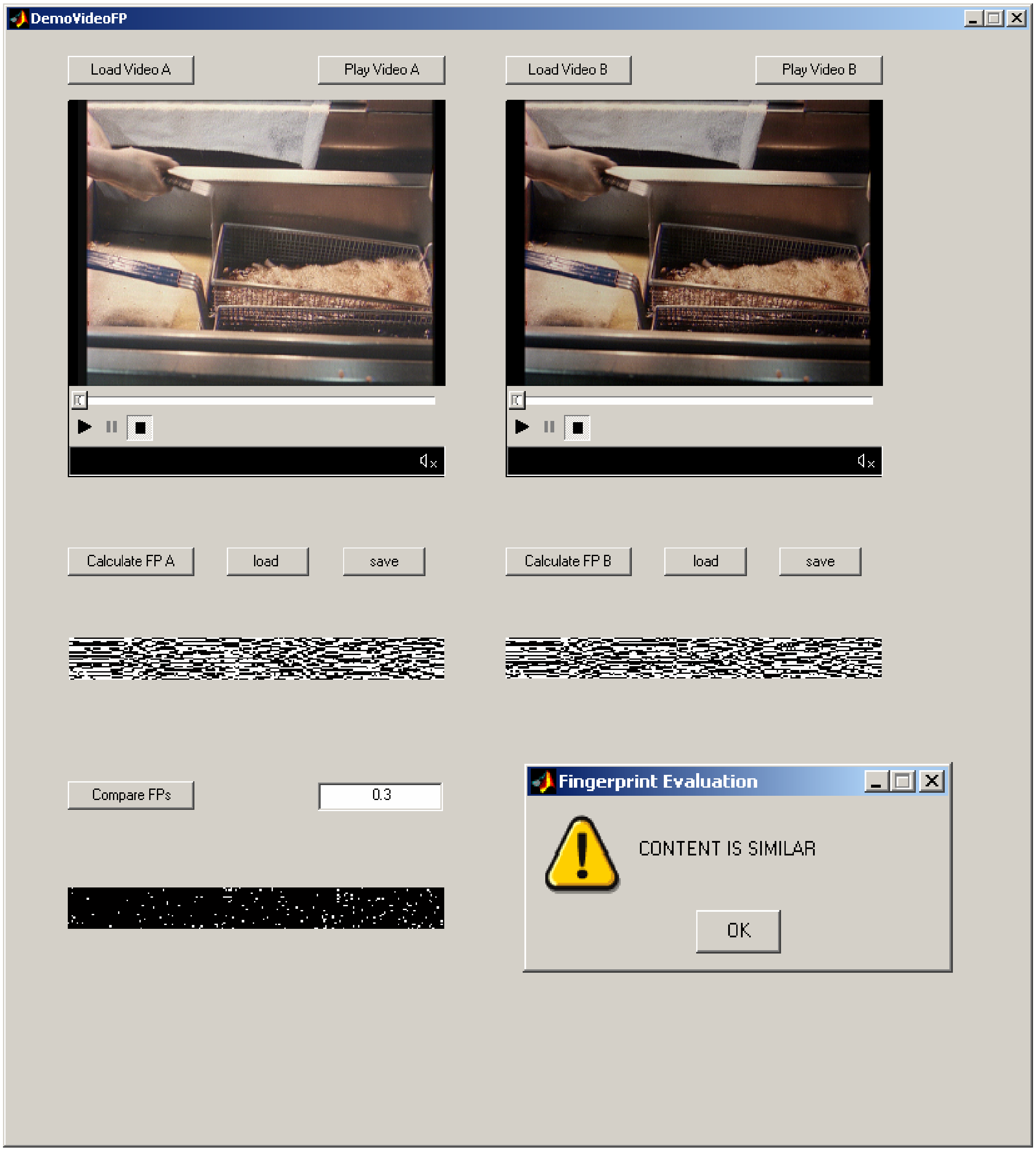
- Identification of suitable features for the calculation of video fingerprints: The basic feature is the similarity between different blocks in consecutive frames. This temporal similarity sequences is chosen due to is discriminability.

- Development and implementation of a video fingerprinting algorithm based on the temporal block similarity sequence as shown in the next figure.



- First analysis of the implemented algorithm: Based on the some internal test images the developed algorithm was analysed and its performance has been successfully compared with other state-of-the-art algorithms.

- The implemented prototype functionality was integrated in a Graphical User Interface (GUI). This GUI allows an simple demonstration of the capabilities of the implemented algorithm.



CONTENT IS SIMILAR

12.4 References

[Atrey2002] P. K. Atrey, W.-Q. Yan, E.-C. Chang, and M. S. Kankanhalli, “A hierarchical signature scheme for robust video authentication using secret sharing,” *Multimedia Modelling Conference*, 2004.

[Caspi2004] Caspi, Y. Bargeron, D.: Sharing video annotations, ICIP '04. 2004 International Conference on Image Processing, 2004

[Cheung2002] S. Cheung and A. Zakhor, “Efficient video similarity measurement with video signature,” *IEEE Trans. on CSVT*, vol. 13, no. 1, pp. 59–74, Jan. 03.

[Gargi1998] U. Gargi, S. Antani, and R. Kasturi, “Performance Characterization and Comparison of Video Indexing Algorithms,” *In CVPR*, Santa Barbara, CA, 1998, pp. 559-565.

[Hanj2002] A. Hanjalic, “Shot-boundary detection:unraveled and resolved?,” *IEEE Trans. on CSVT*, vol. 12, no. 2, Feb. 02.

[Hamp2001] A. Hampapur and R. Bolle, “Comparison of Distance Measures for Video Copy Detection”, IBM Research Report, May 2001.

[Oost2001] J. Oostveen, A. Kalker, and J. Haitsma, "Visual Hashing of Digital Video: Applications and Techniques", *Proc. of SPIE applications of digital image processing 24*, July 2001, San Diego, USA.

[Oost2002] J. Oostveen, T. Kalker, and J. Haitsma, “Visual hashing of digital video: application and techniques,” *SPIE*, 01.

13 Text Fingerprints

13.1 State of the Art

Fingerprint estimation for textual contents aims to compare texts and estimate similarities among them. While the detection of exact copies through the comparison of the whole document checksums is simple and reliable, the detection of partial modifications is an harder task. Another point is to distinguish between fingerprinting procedures that: (a) are able to detect *resemblance* among documents; (b) are able to detect *containment* relationships among documents (Broder 1997).
As pointed out by Heintze (1996), the major topics to be considered in fingerprint estimation are:

- **stored fingerprint / processed fingerprint**. To the aim of comparing fingerprint values, we have to take in account a pair of fingerprints: the one extracted from the original file (and, in case, stored in a repository), the other extracted from the processed file that we have to compare with the original;
- **degree of similarity among documents**. The fingerprint must be usable to estimate the similarity (and containment) relations among documents, through a gradable scale;
- **fingerprint size**. The algorithm must take in account the fingerprints file-size with respect to the granularity of the analysis: since the fingerprint value derives from an hashing procedure on the textual information, the finer is the grain of the hash, the bigger is the size of the fingerprint file; it is needed to get a balance between these two opposite requirements;
- **fingerprint matching**. It involves the structure of the algorithm that match similarities among different fingerprints: linear comparison vs. cross comparison (with respect to the document structure and sub parts);
- **full fingerprint / selective fingerprint**. This topic is related with the two previous ones: calculating the full fingerprint of a document implies to get a large-size extracted file; the extraction of a selective fingerprint deal to the possibility of fine-grained matching task among documents (and their sub-parts).

13.1.1 General requirements

The main requirement to be satisfied by a fingerprint extractor is the independence from the file format. The main technique to solve this problem is to convert all the file formats to ASCII files. Reliable format-conversion libraries are required to assure us to get the same fingerprint value from the same text content given in different format.

As it emerges form the state of the art, fingerprint and copy-detection algorithms should satisfy these three further general requirements:

1. *White space / punctuation insensitivity:* In matching text files, matches should be unaffected by such things as extra white space, capitalization, punctuation, etc. this requirement may deal with some domain specific requirements of insensitiveness (for example, in matching software text it is desirable to make matching insensitive to variable names). Samples of treatment: white space and punctuation are removed,; all letters are converted to lower case; all variable names are replaced by one string

2. *Noise suppression:* Discovering short matches, such as the fact that the word *the* appears in two different documents, is uninteresting. Any match must be large enough to imply that the material has been copied and is not simply a common word or phrase of the language in which documents are written. This requirement deals with the choose of the k length of the k-grams (it has to be longer, in terms of number of characters, than the most common locution phrases).

3. *Position independence:* coarse-grained permutation of the contents of a document (e.g., scrambling the order of paragraphs) should not affect the set of discovered matches. Adding to a
document should not affect the set of matches in the original portion of the new document. Removing part of a document should not affect the set of matches in the portion that remains.

K-gram techniques

Most previous techniques used to extract fingerprints and detect partial copies make use of the *k-grams*, i.e. contiguous sub-strings of length k. the process require to divide a document into k-grams, and extract an hash value from each one. The result of this process is a fingerprint that represent the document in each of its sub-parts of length k.
As Karp and Rabin's (1987) algorithm for fast sub-string matching is apparently the earliest version of fingerprinting based on k-grams, the first scheme to apply fingerprinting to collections of documents was developed by Manber (1994). He exploited Karp-Rabin string matching and applied it to detecting similar files in file systems. Rather than having a single candidate string to search for, in this problem the aim is to compare all pairs of k-grams in the collection of documents.
As a matter of fact, the use of k-grams involves the creation of hashes value for each pattern of length k. In a document, there are almost as many k-grams as characters, as every position in the document marks the beginning of a k-gram (except for the last $k - 1$ positions).

Since there is a k-gram for every byte of an ASCII file, and at least 4-byte hashes are needed for most interesting data sets, a naive scheme that selected all hashed k-grams would create an index much larger than the original documents. This is impractical for large document sets, and the obvious next step is to select some subset of the hashes to represent each document.

Various techniques have been proposed to solve the problem of the fingerprint size and storage. The accuracy of the document fingerprint and of the comparison among different document fingerprints depends on the size-reduction parameters.
Heintze (1996) proposed to choose the n smallest hashes of all k-grams of a document as the fingerprints of that document. By fixing the number of hashes per document, the system would be more scalable as large documents have the same number of fingerprints as small documents.
The price for a fixed-size fingerprint set is that only near-copies of entire documents could be detected. Documents of vastly different size could not be meaningfully compared, with the lack of detection of *containment* relationship among documents.
In Schleimer, Wilkerson and Aiken (2003) the authors propose an innovative method (the *winnowing* algorithm) to solve the problem of fingerprint size and k-grams selection.

Other techniques in comparing document

Instead of using *k*-grams, the strings to fingerprint can be chosen by looking for sentences or paragraphs, or by choosing fixed-length strings that begin with "anchor" words [Brin et al.]. Recently, a system for document similarity detection has develop by the DANCER team, based on a two stage architecture (step 1: chunk similarities detection; step 2: document similarity detection) with a specific a linguistic insight within the analysis (Mandreoli et al. 2003).

13.1.2 Document comparison and plagiarism detection initiatives

Turnitin

http://www.turnitin.com/static/home.html
Recognized worldwide as the standard in online plagiarism prevention, Turnitin helps educators and students take full advantage of the Internet's educational potential. Used by thousands of institutions in over fifty countries, Turnitin's products promote originality in student work, improve student writing and research skills, encourage collaborative learning, and save valuable instructor time. The system is based on the *document source analysis* techniques developed by a team of researcher at UC Berkley.

Koala

http://www-2.cs.cmu.edu/afs/cs/user/nch/www/koala-info.html
Most pieces of research writing evolve over a period of time. Different versions of document are published or otherwise made available at various stages along this process. The purpose of the KDF system is to trace the history of a document by identifying common segments of text between documents. Typical uses of the system include:

- **finding citations:** you have an draft version of a document and want to know where it was eventually published
- **tracing references:** you have a conference abstract and want to know if there is a revised or journal version of the article.
- **tracing plagiarism:** you have a paper and want to check whether it has been previously published in whole or part
    - by the same authors.
    - by different authors.

Document searches proceed in a number of steps. First the document (URL) is loaded by the KDF server. If necessary, the document is converted to a textual representation. Then, using this text, a fingerprint of the document is generated. Finally, this fingerprint is matched against the current document fingerprint database to find related documents.

WCopyFind

http://plagiarism.phys.virginia.edu/Wsoftware.html
This opensource program examines a collection of document files. It extracts the text portions of those documents and looks through them for matching words in phrases of a specified minimum length. When it finds two files that share enough words in those phrases, WCopyfind generates html report files. These reports contain the document text with the matching phrases underlined.

1. WCopyfind 2.4 can compare news services for shared headlines or content.
2. WCopyfind 2.4 can find which words are most common in the news.
3. WCopyfind 2.4 can look for lobbiest language appearing directly in finished legislation.
4. WCopyfind 2.4 can list and count the matching words between two documents.

It cannot search the web or internet to find matching documents for you. You must specify which documents it compares. Those documents can be local ones—on your computer or a file server—or, with versions 2.1 and higher, web-resident documents that are pointed to by local internet shortcuts. If you suspect that a particular web page has been copied, you must create an internet shortcut to that page and include this shortcut in the collection of documents that you give to WCopyfind.
Sourcecodes given in Microsoft Visual C++.Net language.

EVE2 (Essay Verification Engine)

http://www.canexus.com/eve/index.shtml

Eve performs a large number of complex searches to find material from any Internet site. While it would be technically impossible with today's technology to build a program that could check every web site on the entire Internet, EVE2 comes as close as possible by employing the most advanced searching tools available to locate suspect sites. Not only does it find these suspect sites, but it then does a direct comparison of the submitted essay to the text appearing on the suspect site. If it finds evidence of plagiarism, the URL is recorded. Once the search has completed, the teacher is given a full report on each paper that contained plagiarism, including the percent of the essay plagiarized, and an annotated copy of the paper showing all plagiarism highlighted in red.

Moss

http://www.cs.berkeley.edu/~aiken/moss.html

Moss (for a Measure Of Software Similarity) is an automatic system for determining the similarity of C, C++, Java, Pascal, Ada, ML, Lisp, or Scheme programs. To date, the main application of Moss has been in detecting plagiarism in programming classes. Since its development in 1994, Moss has been very effective in this role. The algorithm behind Moss is a significant improvement over other cheating detection algorithms (at least, over those known to us).

13.2 Problems

Fingerprint for the content of a textual file is at present a puzzling task. The state of the art does not provide us even with a reliable procedure to ensure the exact matching of fingerprints derived from documents with same content in different format, that in principle should be the easiest case.

Solution:

Given this, the strategy of document comparison must ensure at least the possibility to evaluate the degree of similarity between texts.

In the AXMEDIS framework, similarity evaluation is a method enabling the identification of plagiarism at the textual level. Through the Content fingerprint and similarity estimation a provider can assess the degree of similarity of a given texts with those stored in the repository.

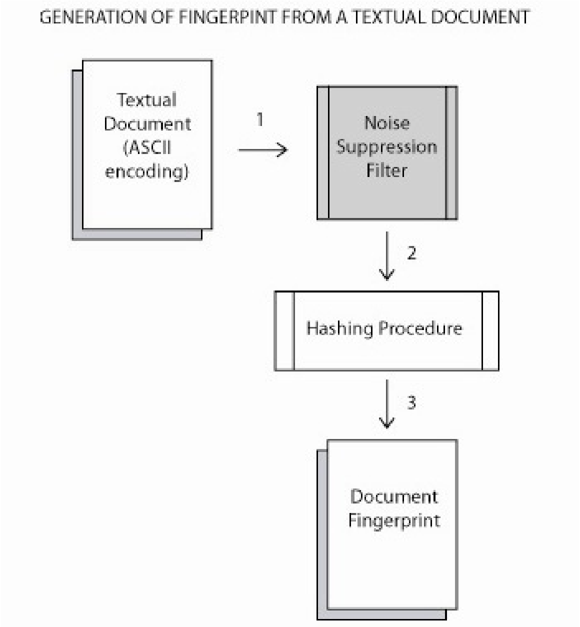
13.3 Work Performed

On the basis of literature algorithms, the following strategies have been designed:

A) Generation of fingerprint from a textual document

Given a plain text file (ASCII encoding), the content fingerprint is generated through the following steps:

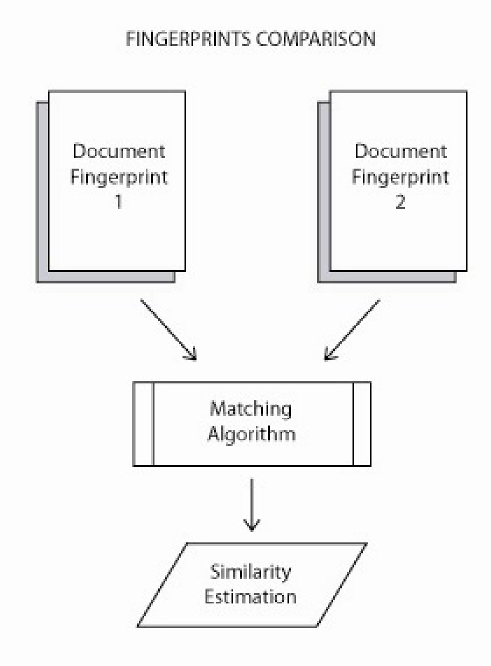
1. Noise Suppression Filter (pre-processing that aims to eliminate or convert problematic characters: suppression of blank spaces, paragraphs, punctuation, characters out of the ASCII-standard encoding); vowels may be eliminated, taking into account only consonant sequences; all letters are converted to lower case.
2. Hashing Procedure: k-gram hashing algorithm (dimension of k-grams has to be over a certain threshold, that ensure to consider significant portion of text, over the common locution phrases). Reliable hashing algorithms to be taken into account for this purpose are described in Heintze 1996 and Schleimer et el. 2003.
3. Document Fingerprint: output of the hash.


GENERATION OF FINGERPINT FROM A TEXTUAL DOCUMENT
Textual Document (ASCII encoding)
1
Noise Suppression Filter
2
Hashing Procedure
3
Document Fingerprint

B) Fingerprints comparison

The matching process output the similarity estimation between two different hash values. The estimation is run over the values and searches similar hash sequences which appear in both fingerprints (document fingerprint 1 and document fingerprint 2; see, for example, WCopyFind output of similarities among documents).

Similarity Estimation: similarity degree among documents is determined over a scale of percentage with respect to the document fingeprint l, taking into account the character sequences contained in the document fingerpint 1 that are similar to the ones in the document fingerpint 2. The more general cases regarding relationship between documents are (see Broder 1997):
- resemblance relationship: the document 1 is similar (in a certain percentage) to document 2;
- containment relationship, distinguished into two sub-cases:
    (a) subset relationship (the document 1 is similar to a part of document 2);
    (b) container relationship (the document 1 contains the document 2, or parts of it).
In all cases, given that the degree of similarity is computed on the document 1, the result is the percentage of similarity between documents that gives an estimation on how much the content of the document 1 file is composed by document 2 materials.



13.4 References

Brin, S., Davis, J., García Molina, H. (1995), Copy detection mechanism for digital documents. In *Proceedings of the ACM SIGMOD Conference, 1995*.

Broder, A. (1997), On the resemblance and containment of documents. In *SEQS: Sequences 91*.

Heintze, N. (1996), Scalable document fingerprinting. In *1996 USENIX Workshop on Electronic Commerce*.

Karp, R. M., Rabin M. O. (1987), Pattern-matching algorithms. *IBM Journal of Research and Development, 31(2)*.

Manber, U.(1994), Finding similar files in a large file system. In *Proceedings of the USENIX Winter 1994 Technical Conference*.

Mandreoli, F., Martoglia, R. Tiberio, P. (2003), Un metodo per il riconoscimento di duplicati in collezioni di documenti.

Schleimer, S., Wilkerson, D. S., Aiken, A. (2003), Windowing: Local Algorithms for Document Fingerprinting. In *Proceedings of the ACM SIGMOD Conference, 2003*.

14 Any Digital Media Types

14.1 State of the Art

14.1.1 MD5

The MD5 algorithm is a cryptographic *one-way hash function* which means that it operates on an arbitrary-length message and returns a fixed-length *hash value*, sometimes called the *message digest* (therefore MD) which is a condensed representation of the *hashed* message and can hence be looked at as a fingerprint of that message.

MD5 is an improved version of MD4, designed by Ronald L Rivest of MIT. Since some parts of the MD4 algorithm had successfully been attacked Rivest strengthend MD4 resulting in MD5. Although more complex than MD4, it is similar in design and also produces a 128-bit hash value.

In general, the one-wayness of one-way hash functions is realized by using a *compression function* which is itself a one-way function. It outputs a hash value of length n given an input of some larger length m, namely a block of a message and the ouput of the previous call of the compression function (last hash value). That is, the message is divided into blocks which are then sequentially processed by using the compression function of the algorithm. The result of each call of this function is then the *updated hash* which is the hash value of all blocks of the message up to that point. In case the length of the message is not a multiple of the block length, the message has to be padded by appending bits in a prespecified way. The message is then referred to as *padded message* while messages before any addition of padding bits are referred to as *original messages*.

In general, cryptographic one-way hash functions have the following characteristics:

1. Given a message, it is easy to compute its hash value.
2. Given a hash value it is hard to compute the message.
3. Given a message, it is hard to find another message which hashes to the same value.
4. It is hard to find two random messages which hash to the same value.

The last point is not a characteristic of one-wayness, but it is an absolute requirement, when the hash function is used along with a signature algorithm to compute a digital signature. It is called *collision-resistance*. A hash algorithm which meets all requirements listed here may then be called *secure* or a *cryptographic* one-way hash function. (See [2] for more information about how a birthday attack can successfully be applied on a hash algorithm which is not collision-resistant.) Since MD5 is intended to be used in digital signature applications, it requires to be collision-resistant. In [1] this requirement is claimed to be met by the algorithm.

It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. 1em — [1]

This is point 4 and 3 respectively of the characteristics mentioned above. Here it is *conjectured* that the algorithm is collision-resistant since this property depends first on the length of the computed hash value in bits (which should be sufficient even today since it is$128\,bit$) and second, the security of the compression function since it is assumed that the hash algorithm is secure as long as there is no known successful attack against its compression function. Therefore the conjecture was ok at the time the RFC was written. But unfortunately, meanwhile the compression function of MD5 can no longer be considered as being collision-resistant.

A more successful attack by den Boer and Bosselaers produces collisions using the compression function in MD5. ... It does mean that one of the basic design principles of MD5 – to design a collision-resistant compression function – has been violated. — [2]

Although some people assume that this weakness of the compression function has *no practical impact* on the security of MD5, hash algorithms like SHA-1 should be used instead of MD5 whenever possible. Additionaly the SHA-1 provides a security of $2^80\,bits$ against birthday attacks.

14.1.2 SHA-1

NIST, along with the NSA, designed the Secure Hash Algorithm (SHA/SHA-1) which is used in the Secure Hash Standard (SHS). The SHA-1 is specified in FIPS PUB 180-1 ([3]).

According to this publication:

This standard specifies a Secure Hash Algorithm (SHA-1) which can be used to generate a condensed representation of a message called a message digest. The SHA-1 is required for use with the Digital Signature Algorithm (DSA) as specified in the Digital Signature Standard (DSS) and whenever a secure hash algorithm is required for federal applications. The SHA-1 is used by both the transmitter and intended receiver of a message in computing and verifying a digital signature.— [3]

Although this module is named *C_SHA*, it is an implementation of the SHA-1 algorithm which is an updated version of the original SHA. Both, the SHA and the SHA-1 are cryptographic one-way hash functions which means that they operate on an arbitrary-length message and return a fixed-length *hash value*, sometimes called the *message digest*. Additionally, one-way hash functions have the following characteristics:

5. Given a message, it is easy to compute its hash value.
6. Given a hash value it is hard to compute the message.
7. Given a message, it is hard to find another message which hashes to the same value.
8. It is hard to find two random messages which hash to the same value.

The last point is not a characteristic of one-wayness, but it is a very useful characteristic of any hash function. It is called *collision-resistance*. Collision-resistance is an absolute requirement, when the hash function is used along with a signature algorithm to compute a digital signature. See [2] for more information.

These requirements given by point one to four are met by the SHA-1 algorithm:

The SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify. — [3]

The standard of the SHA-1 algorithm, the SHS also explains why the message digest is used by the DSA algorithm rather then the original message. It also specifies the maximum length of that message and the length of its message digest.

When a message of any $length < 2^{64}$ *bits* *is input, the SHA-1 produces a 160-bit output called a message digest. The message digest can then be input to the Digital Signature Algorithm (DSA) which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message. The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature.* — [3]

14.2 Problems

In the last months, several new attacks on cryptographic hashing algorithms have been published. These attacks have strong implications on the security of the available cryptographic hashing algorithms. As a consequence existing risk analyses have to be revised carefully and consider recent attacks.

For MD5-hash functions Xiao Yun Wang's discovery [Wang04] proves that in theory some commonly used encryption algorithms including MD5 are not secure. Similarly, attacks on SHA-1 from the same author together with other colleagues showed also weaknesses in this commonly used cryptographic hash algorithm. Although research is looking for alternatives, a short-term solution is not in sight.

SHA-1 is nevertheless considered to be valid till 2009. Within AXMEDIS, new developments and insights have to be considered carefully. Due to the modular design and the plug-in interface new algorithms can be integrated easily.

14.3 Work performed

So far, the first algorithms have been integrated and tested extensively using regression tests, including:

../Framework/source/fingerprint/general_resource/crypto/AXM_Digest/AXM_Digest.c
../Framework/source/fingerprint/general_resource/crypto/AXM_MD5/AXM_MD5.c
../Framework/source/fingerprint/general_resource/crypto/AXM_SHA/AXM_SHA.c
../Framework/source/fingerprint/general_resource/tests/crypto/AXM_Digest/AXM_TestDigest.h
../Framework/source/fingerprint/general_resource/tests/crypto/AXM_Digest/AXM_TestDigest_ErrorsHarnesses.h
../Framework/source/fingerprint/general_resource/tests/crypto/AXM_Digest/AXM_TestDigest_HashFile.c
../Framework/source/fingerprint/general_resource/tests/crypto/AXM_Digest/AXM_TestDigest_PreCondVfctn.c
../Framework/source/fingerprint/general_resource/tests/crypto/AXM_MD5/AXM_ErrorsTestMD5.h
../Framework/source/fingerprint/general_resource/tests/crypto/AXM_MD5/AXM_TestMD5.h

../Framework/source/fingerprint/general_resource/tests/crypto/AXM_MD5/AXM_TestMD5_TestAll.c
../Framework/source/fingerprint/general_resource/tests/tools/AXM_ToolLib/AXM_AsciiBinary_Test.c
../Framework/source/fingerprint/general_resource/tests/tools/AXM_ToolLib/AXM_AsciiBinary_Test.h
../Framework/source/fingerprint/general_resource/tools/AXM_TestOutput/AXM_TestOutput.c
../Framework/source/fingerprint/general_resource/tools/AXM_ToolLib/AXM_AsciiBinary.c
../Framework/source/fingerprint/general_resource/tools/AXM_ToolLib/AXM_Base64.c
../Framework/source/fingerprint/general_resource/tools/AXM_ToolLib/AXM_CmdLine.c
../Framework/source/fingerprint/general_resource/tools/AXM_ToolLib/AXM_EnDeCode.c
../Framework/source/fingerprint/general_resource/tools/AXM_ToolLib/AXM_Error.c
../Framework/source/fingerprint/general_resource/tools/AXM_ToolLib/AXM_File.c
../Framework/source/fingerprint/general_resource/tools/AXM_ToolLib/AXM_Trace.c

The complete source code is available in the AXMEDIS version control system (see above paths).

14.4 References

[1] Ronald L. Rivest. *RFC 1321: The MD5 Message-Digest Algorithm*. The Internet Engineering Task Force (IETF), Status: Informational edition, April 1992.
[2] Bruce Schneier. *Applied Cryptography (Second Edition)*. John Wiley & Sons, Second Edition edition, 1996.
[3] National Institute of Standards and Technology. *FIPS PUB 180-1: Secure Hash Standard*. Gaithersburg,

15 Query Support in the AXMEDIS Database

The query support is a combination of query mechanisms for retrieving content objects among the indexed information, in the AXMEDIS database (for content processing, composition, formatting, etc.), on the P2P AXEPTool, on the content collector referencing the content contained in the CMSs (Crawled Results Integrated Database). From the AXMEDIS database interface it has to be possible to make valid queries for all the environments and read unified results. From the AXEPTool or from the Collector the queries can be performed only on their environment.
The query support allows the specification of technical/professional query including metadata, technical information, business and licensing aspects, content based, DRM rules, etc.
The query Support Web Service Interface communicate with the backend of the AXDB for gathering information directly on the object stored in the AXMEDIS database, while access via a web service interface to the crawling system and to the AXEPTOOL. The web service interface and therefore the WSDL offered by the AXEPTOOL and by the Crawling system must be the same as that offered by Query Support in order to have a unique format for data exchanging.
The Query User Interface access via web service to the Query support in order to send query and gather information.

15.1 State of the Art

There is nothing in the literature about this model since it has been developed specifically for AXMEDIS.
The state of the art is discussed directly in this document commenting the Query Support for the AXDB.

15.2 Problems

The main problems related to the translation of a query on the AXDB are related to the presence of optional fields and of translation of fields that compel to have a very complex structure in terms of equivalent SQL query.
Another principal problem is related to the complexity of the resulting query that contains a lot of JOINS over all the tables that are considered in the query.
The third aspect to be considered is related to the distribution of queries towards the database of licenses and PAR and the related unification of results.

15.3 Work Performed

The work performed is mainly related to the solution of the cited problems in terms of automatic translation of queries minimizing the number of joins and using inner joins instead of outer joins to minimize the complexity of the resulting query.
The other aspects are mainly related to the mapping of metadata and to the definition of a query language independent of the different languages that can be adopted inside the project.

15.3.1 General overview of the work performed

In the following a general description of the work performed in this period has been summarized together with the percentages of completition.

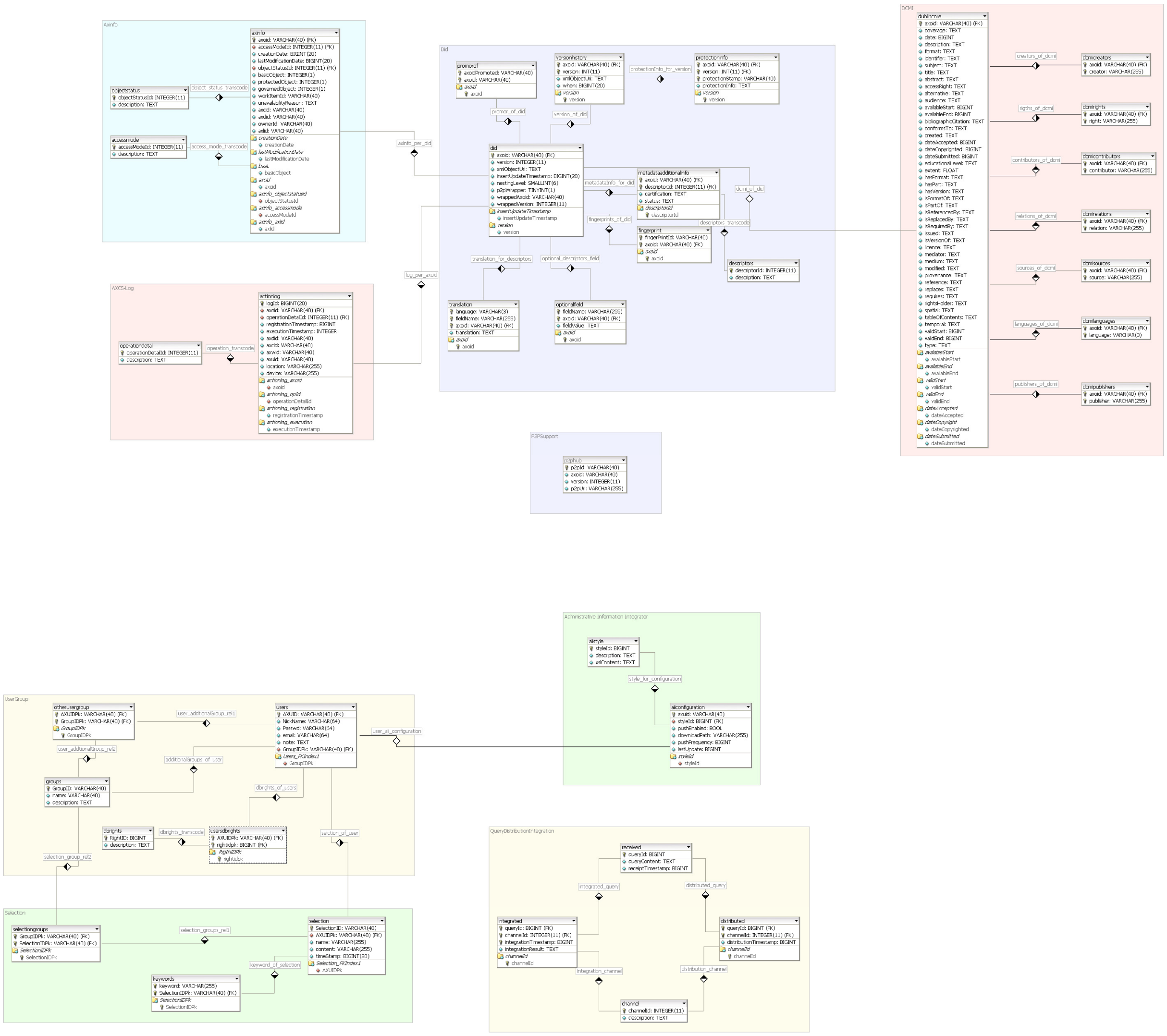
- AXMEDIS OBJECTS REPOSITORY RELATION SCHEMA
  - OK 100%
- DESCRIPTORS METADATA MAPPING IN RELATIONAL SCHEMA
  - OK 100%
- ACCOUNT LOG FOR AXMEDIS OBJECTS REPOSITORY
  - OK 100%
- DATABASE SCHEMA FOR SUPPORTING AXMEDIS
  - OK 100%
- AXMEDIS DATABASE INTERFACE
  - OK, 1st prototype ready with 80% of functionalities. Main missing functionalities are the missing functions for Axdb administrative, for Query result collecting (async only) and some of those related to CAMART and AII.
- AXMEDIS DATABASE WEBSERVICE INTERFACE
  - Descriptor_Support
    - OK, 1st prototype ready (100%)
  - Publication_Support
    - OK, 1st prototype ready (100%)
  - User_Support
    - OK, 1st prototype ready (100%)
  - AXDBQuerySupport
    - OK, 1st prototype ready (sync communication 100%, async 80%, also if not implemented yet, but basic functionalities are the same)
  - P2PHubNode
    - OK, 1st prototype ready (100%)
- AXMEDIS ADMINISTRATIVE WEB DATABASE INTERFACE
  - OK, 1st prototype ready (some minor missing functionalities, for detailed authentication, but now the underlying layer is ready and for meliorating user interface usability
- AXMEDIS OBJECT LOADER/SAVER
  - Loader:
    - OK, 1st prototype (90%, if we want to implement some more reliable mechanism for transferring objects not based on their physical location, otherwise 100%. It has to be discussed since it is not easy to move large objects)
- HISTORY OF AXMEDIS OBJECTS EVOLUTION IN PRODUCTION REPOSITORY (EXITECH, FUPF)
  - OK, 1st prototype done. Waiting for some real object and some sample of the size (some hints received by ILABS) in order to choose technologies and to completed with real tests. The missing Saver (due to delayed AXOM) is another reason for having not completed this part.
- QUERY AND SELECTION ARCHIVE FOR EACH USER (EXITECH, DSI, HP, XIM, IRC)
  - OK, 1st prototype ready (Little delay, 60% Some missing functionalities at Web service level. Implemented first versions of loadSelection, deleteSelection, saveSelection).
- QUERY SUPPORT WEB SERVICE INTERFACE (EXITECH)

    - OK, 1st prototype ready (sync communication 100%, async 80%, also if not implemented yet, but basic functionalities are the same)
- QUERY DISTRIBUTION (EXITECH)
    - OK, 1st prototype ready (sync communication 100%, async 80%, also if not implemented yet, but basic functionalities are the same)
- QUERY RESULTS INTEGRATION (EXITECH)
    - OK, 1st prototype ready (sync communication 100%, async 60%)

15.3.2 Data model incl. meta data selection and mapping

Data model has been implemented in AXMEDIS by adopting a new structure for the database schema that is reported below and is also detailed in the revised version of the specification DE 3.1.2E Database-Gathering version 2.3 or higher.

The following schema is the AXDB schema really implemented and running under the services that will be detailed in the following.



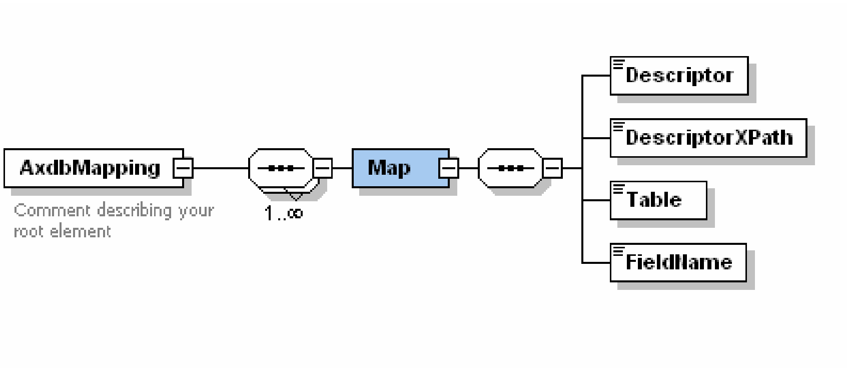
Over this database structure some major functionalities have been implemented:

- **authenticateUser**: verifies if a user is authenticated with a password for performing a certain operation
- **deleteObjectVersion**:allows the elimination of one version or all versions of an object
- **getAxobj**: allows to obtain from an axoid, the Did representation of an object.
- **getModifiedAxoidAfter**: returns a list of AXOIDs that have been modified after a certain timestamp given as input parameter.
- **getObjectURI**: returns the URI of the object/version passed as in parameters
- **getUserIdFromNick**: returns the userId (axuid) associated to the given nickname
- **listAvailableVersion**: allows to get th numbers of all the available versions of an object
- **setUnavailable**: sets an object as unavailable.

Other functionalities implemented are related to the web services that are related to the database. Implemented web services, which details can be found in the specification document DE 3.1.2E Database-Gathering version 2.3 or higher are:

- Descriptor_Support
- Publication_Support
- User_Support
- P2PHubNode
- Loader Web service has been implemented as a first prototype with both sync and async functionalities
- QUERY AND SELECTION ARCHIVE FOR EACH USER: loadSelection, deleteSelection, saveSelection functionalities have been implemented

The main aspect related to the metadata mapping is related to the fact that each company can have inside the factory a different mapping of the metadata of the AXMEDIS object inside the AXDB. This fact compels us to add a mapping mechanism to associate each database field to an Xpath inside the AXMEDIS object. The same Xpath must be present in the queries in order to identify univocally the desired field.
The data model identified
The format proposed for the mapping has to consider the descriptor where the record is (only for reference), the full path of the item to be found inside the object and the name of the table and of the field that is wanted in the AXDB.
In the case of OptionalFields, the fieldname will be used as a record of the table as for all the other 1 to n relationship in ER model.
The schema for this representation is reported in the following schema, using XPath for describing the fullPath in the DecriptorXPath tag:



The textual version of the schema follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
        <xs:element name="AxdbMapping">
                <xs:annotation>
                        <xs:documentation>Comment describing your root element</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                        <xs:sequence maxOccurs="unbounded">
                                <xs:element name="Map">
                                        <xs:complexType>
                                                <xs:sequence>
                                                        <xs:element name="Descriptor" type="xs:string"/>
                                                        <xs:element name="DescriptorXPath" type="xs:string"/>
                                                        <xs:element name="Table" type="xs:string"/>
                                                        <xs:element name="FieldName" type="xs:string"/>
                                                </xs:sequence>
                                        </xs:complexType>
                                </xs:element>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

In the following example file, it is reported an example of one of the fields in the mpeg7 descriptor mapped to the field Spatial of DublinCore (in has no sense, it is only for giving an example) .

```
<?xml version="1.0" encoding="UTF-8"?>
<AxdbMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="axdb-mapping-v-
1-2.xsd">
        <Map>
                <Descriptor>dsc_mpeg7</Descriptor>
                <DescriptorXPath>/Statement/Mpeg7/DecsriptionUnit/MediaFormat/BitRate</DescriptorXPath>
                <Table>DublinCore</Table>
                <FieldName>Spatial</FieldName>
        </Map>
</AxdbMapping>
```

In the following other example the final part of the path is an attribute of a tag instead of a and it is mapped to an OptionalFiled record whose field name is visualcodingformathref**.**

```
<?xml version="1.0" encoding="UTF-8"?>
<AxdbMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="axdb-mapping-v-
1-2.xsd">
        <Map>
                <Descriptor>dsc_mpeg7</Descriptor>

        <DescriptorXPath>/Statement/Mpeg7/DescriptionUnit/MediaFormat/VisualCodingFormat[@href]</DescriptorXPath>
                <Table>OptionalField</Table>
                <FieldName>visualcodingformathref</FieldName>
        </Map>
</AxdbMapping>
```

15.3.3 Query description language formalisation (including metadata, technical information, business and licensing aspects, content based, DRM rules, etc)

In this section the general schema for representing an AXMEDIS query is reported and commented.
The schema of the query is mainly composed by three parts. In the first part the location on which the query has to be distributed is reported, in the second parts a list of fields that the query should give as the result is contained and in the third and final part, the conditions under which a query has to be executed are imposed.

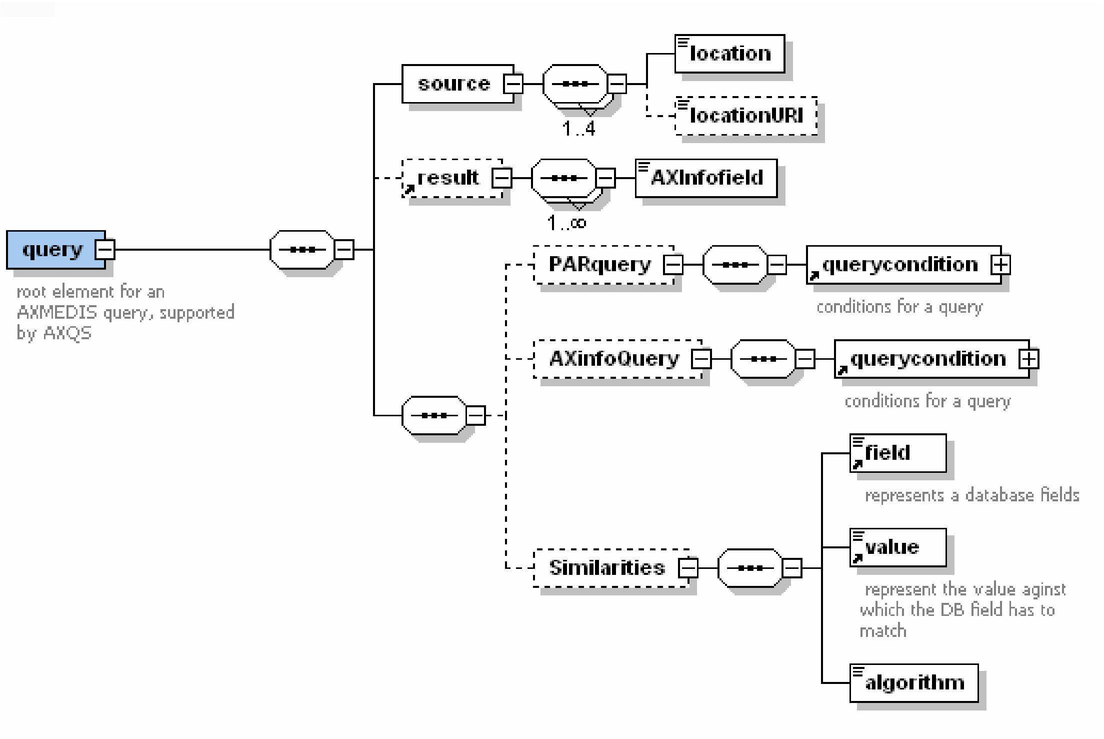
The three different parts are separately commented after the textual and graphic representation of the Query schema in sections

The proposed schema is reported below, in text format and in a more fashionable graphic format that cen be useful to have a general overview.

According to the last revision of the query model and of the way in which query can be submitted by the user, it can be noted that user performs queries on Descriptors and PAR/InternalPar for obtaining a list of results that satisfy the criteria. The DRM part is managed in a second step of the query, where from the licence DB, all the user that have a licence to sell a licence to the user according to the PAR limitation imposed, are automatically extracted, so that the user the performed the query can select the most appropriate to its needs. For these reasons the limitation on DRM that were present in the previous query model have been removed, moving this functionality to the automatic generation of a list of possible licensors.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="result">
        <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
                <xs:element name="AXInfofield" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="query">
        <xs:annotation>
            <xs:documentation>root element for an AXMEDIS query, supported by AXQS</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="source">
                    <xs:complexType>
                        <xs:sequence maxOccurs="4">
                            <xs:element name="location">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:enumeration
value="CRAWLER"/>
                                        <xs:enumeration
value="AXEPTOOL"/>
                                        <xs:enumeration value="AXDB"/>
                                        <xs:enumeration value="QS"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="locationURI" type="xs:anyURI"
minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element ref="result" minOccurs="0"/>
                <xs:sequence>
                    <xs:element name="PARquery" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="querycondition"/>
                            </xs:sequence>
                            <xs:attribute name="InternalPAR" type="xs:boolean"
use="optional"/>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="AXinfoQuery" minOccurs="0">
                        <xs:complexType>
```

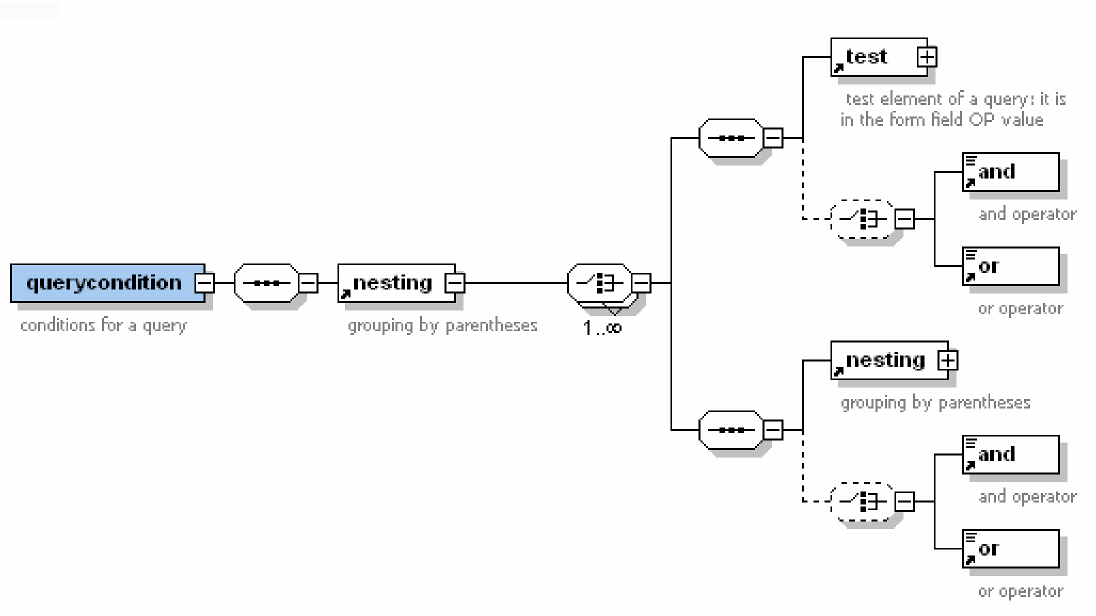
```
                            <xs:sequence>
                                <xs:element ref="querycondition"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="Similarities" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="field"/>
                                <xs:element ref="value"/>
                                <xs:element name="algorithm">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration
value="REGULAREXPRESSION"/>
                                            <xs:enumeration
value=""/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:sequence>
            <xs:attribute name="name" type="xs:string" use="optional" default="QUERY"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="field" type="xs:string">
        <xs:annotation>
            <xs:documentation> represents a database fields</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="value" type="xs:anySimpleType">
        <xs:annotation>
            <xs:documentation> represent the value aginst which the DB field has to match</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="test">
        <xs:annotation>
            <xs:documentation> test element of a query: it is in the form field OP value</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="field"/>
                <xs:element name="operator">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="GT"/>
                            <xs:enumeration value="LT"/>
                            <xs:enumeration value="EQ"/>
                            <xs:enumeration value="GE"/>
                            <xs:enumeration value="LE"/>
                            <xs:enumeration value="NE"/>
                            <xs:enumeration value="STARTWITH"/>
                            <xs:enumeration value="ENDWITH"/>
                            <xs:enumeration value="CONTAINS"/>
                            <xs:enumeration value=""/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element ref="value"/>
            </xs:sequence>
            <xs:attribute name="NOT" type="xs:boolean" use="optional" default="false"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="nesting">
        <xs:annotation>
            <xs:documentation>grouping by parentheses</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:choice maxOccurs="unbounded">
                <xs:sequence>
                    <xs:element ref="test"/>
                    <xs:choice minOccurs="0">
                        <xs:element ref="and"/>
                        <xs:element ref="or"/>
                    </xs:choice>
                </xs:sequence>
                <xs:sequence>
                    <xs:element ref="nesting"/>
                    <xs:choice minOccurs="0">
                        <xs:element ref="and"/>
                        <xs:element ref="or"/>
                    </xs:choice>
                </xs:sequence>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="querycondition">
        <xs:annotation>
            <xs:documentation>conditions for a query</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="nesting"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="and" type="xs:token">
        <xs:annotation>
            <xs:documentation> and operator</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="or" type="xs:token">
        <xs:annotation>
            <xs:documentation> or operator</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:schema>
```



In this first diagram we suppose that a query can be performed on different sources, returning different AXINFO fields and that the query conditions can be expressed on three different information sources:

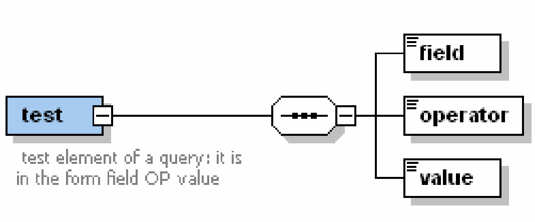
- PARquery, that are the info related to Possible Available Rights
- DRMMquery, that are the information about licensing
- AXinfoQuery, that are the info related to the medatada of the AXMEDIS object.

For each one of these groups a query condition applies and the groups are merged on an AND basis. Querycondition is reported in the following schema:



Querycondition is then a set of nesting elements that represents basically the parentheses levels.

A nesting can be nested in another nesting element of it can be a part of the real query, in the case it is part of a real query it is a combination of test items that are reported in the following:



Test is the basic element where a fields can be compared to a value by the means of an operator that will be discussed in more details in the following.
Test element has also an attribute that allows to specify if it must be negated or not.

15.3.3.1 Query sources

The AXMEDIS query can be distributed on four different locations that are: CRAWLER, AXDB, AXEPTOOL, and other remote query supports. This is the way the source tag has between 1 and 4 locations that can have a value in the enumeration CRAWLER, AXEPTOOL, AXDB, QS and have also an optional URL for covering the need of a remote Query Support that is needed by the Kiosk, when the kiosk will look in the kiosk factory if an object is preseeent or not. This new addition allows also to create a network of query supports that are cooperatively in an hierarchy tree.

15.3.3.2 Query result

The second part of the schema is used to describe the fields that the query should return together with the AXOID (or the temporary AXOID in the case of CRAWLER) that is considered a mandatory field. If the result tag is not present, then only the list of AXOID will be returned; otherwise at least one field must be specified. After the field list an optional array of sorting parameter can be inserter; each sortby tag has an optional attribute (not shown in the graphic version) that by default assumes the asc=true value, that means ascending sorting. If imposed to false, it means that sorting is descending.
The field tag will be limited to the possible list of fields, once the database schema will be defined and a selecting among the possible fields to be queried will be defined.

15.3.3.3 Query conditions

The most relevant part of a query is the section related to query conditions that is of course optional, but if present must have the structure reported in the previous XML schema. All the field tag must contains the database field that must match, all value field must contain the value against which the field is checked and operator is one of the following:

- GT: greater than
- LT: less than
- GE: greater equal
- LE: less equal
- EQ: Equal
- NE: Not equal
- STARTWITH: field must start with the value
- ENDWITH: field must end with the value
- CONTAINS: field must contain in any position the value

The field tag will be limited to the possible list of fields, once the database schema will be defined and a selecting among the possible fields to be queried will be defined.

15.3.3.4 Similarities

Similarities are a particular way of making a specific similarity request on the set ob objects selected by the previous queryconditions. By now only one similarity field as been inserted in order to reduce the complexity of the query. Similarities can be addressed by regular expressions (i.e. for searching a partially known AXOID) or by more advanced algorithms to be defined on the basis of the needs of the project.

15.3.3.5 XML Query example file

In this example the query proposed as a sample by ILABS that is:

SELECT * **FROM** * **WHERE ((**AUTHOR **.EQ.** "BOTTICELLI" **.AND. (** MEDIA **.EQ.** "VIDEO" **.OR.** MEDIA **.EQ.** "AUDIO" **.OR.** MEDIA **.EQ.** "TEXT"**)) .AND.** (IPR **.EQ.** "FREE" **.AND.** COST .LT. "10.00"**))**

Apart from the fields for which we suppose to have only the AXOID and

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="QUERY-v1-6.xsd">
    <source>
        <location>CRAWLER</location>
        <location>AXEPTOOL</location>
        <location>AXDB</location>
    </source>
    <AXinfoQuery>
        <querycondition>
            <nesting>
                <nesting>
                    <nesting>
                            <test>
                                <field>AUTHOR</field>
                                <operator>EQ</operator>
                                <value>BOTTICELLI</value>
                            </test>
                    </nesting>
                    <and/>
                    <nesting>
                            <test>
                                <field>MEDIA</field>
                                <operator>EQ</operator>
                                <value>VIDEO</value>
                            </test>
                            <or/>
                            <test>
                                <field>MEDIA</field>
                                <operator>EQ</operator>
                                <value>AUDIO</value>
                            </test>
                            <or/>
                            <test>
                                <field>MEDIA</field>
                                <operator>EQ</operator>
                                <value>TEXT</value>
                            </test>
                    </nesting>
                </nesting>
                <and/>
                <nesting>
                        <test>
                            <field>IPR</field>
                            <operator>EQ</operator>
                            <value>FREE</value>
                        </test>
                        <and/>
                        <test>
                            <field>COST</field>
                            <operator>LT</operator>
                            <value>10.00</value>
                        </test>
                </nesting>
            </nesting>
```

```
                </querycondition>
        </AXinfoQuery>
</query>
```

Examples can be as complex as we want since the possibility of combining with AND and OR the number of desired nesting levels.
Another sample of a simpler query is reported in the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation=" QUERY-v1-6.xsd">
        <source>
                <location>AXDB</location>
        </source>
        <AXinfoQuery>
                <querycondition>
                        <nesting>
                                <test>
                                        <field>PIPPO</field>
                                        <operator>EQ</operator>
                                        <value>3</value>
                                </test>
                                <and/>
                                <nesting>
                                        <test>
                                                <field>PAPERINO</field>
                                                <operator>EQ</operator>
                                                <value>1</value>
                                        </test>
                                        <or/>
                                        <test>
                                                <field>PLUTO</field>
                                                <operator>GT</operator>
                                                <value>1</value>
                                        </test>
                                </nesting>
                        </nesting>
                </querycondition>
        </AXinfoQuery>
</query>
```

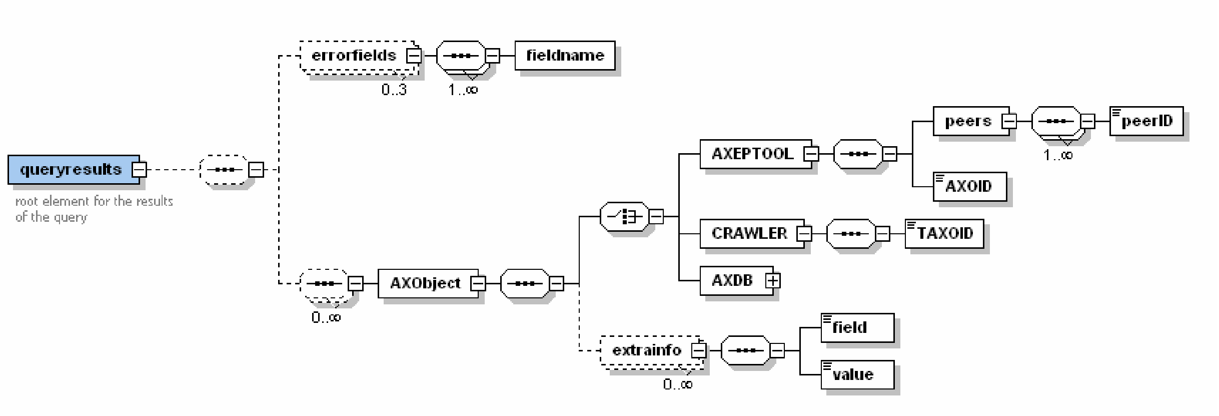
15.3.4 Query and results exchange and mapping (including languages)

In this section the schema for the results that are returned back after a query is reported and discussed.
It is important to track if the result is coming from a source (i.e. CRAWLER, that means that some automatic operation for the creation of an AXMEDIS object starting from the content in the CMS have to be performed before using the object) or from a remote channel such as AXEPTOOL that means that the content is not immediately available, but has to be downloaded from a remote peer.
The information related to the source of information must came together with the fields requested by the user in the query that can be also translations specifically addressed in the AXDB by the translation table. In this way we can

The proposed schema, that an evolution of the schema proposed in the specification, is reported below, in text format and in a more fashionable graphic format that cen be useful to have a general overview.
The evolution has been necessary to address the possibility of having no results from one or more source simply because the fields involved in the query are not present in the AXDB of the company.
The evolution of the schema allows the user to be notified of this fact in order to optionally change the query in order to have a better results set.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2005 sp1 U (http://www.xmlspy.com) by Fabrizio Fioravanti (Exitech) -->
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="queryresults">
        <xs:annotation>
            <xs:documentation>root element for the results of the query</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence minOccurs="0">
                <xs:element name="errorfields" minOccurs="0" maxOccurs="3">
                    <xs:complexType>
                        <xs:sequence maxOccurs="unbounded">
                            <xs:element name="fieldname"/>
                        </xs:sequence>
                        <xs:attribute name="source" use="required">
                            <xs:simpleType>
                                <xs:restriction base="xs:ENTITIES">
                                    <xs:enumeration value="AXEPTOOL"/>
                                    <xs:enumeration value="CRAWLER"/>
                                    <xs:enumeration value="AXDB"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:attribute>
                    </xs:complexType>
                </xs:element>
                <xs:sequence minOccurs="0" maxOccurs="unbounded">
                    <xs:element name="AXObject">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:choice>
                                    <xs:element name="AXEPTOOL">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element
name="peers">

    <xs:complexType>

    <xs:sequence maxOccurs="unbounded">

        <xs:element name="peerID" type="xs:string"/>

    </xs:sequence>

    </xs:complexType>
                                                </xs:element>
                                                <xs:element
name="AXOID" type="xs:string"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="CRAWLER">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element
name="TAXOID" type="xs:string"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="AXDB">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element
name="AXOID" type="xs:string"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:choice>
                                <xs:element name="extrainfo" minOccurs="0"
maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element
name="field" type="xs:string"/>
                                            <xs:element
name="value" type="xs:string"/>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```



The schema of the result is quite simple since it is an optional list of AXMEDIS objects (AXObject) and for each object the information source is reported (AXEPTOOL, CRAWLER, AXDB) together with the information that are relevant for the source, the list of peers that have the object together with the AXIOD for AXEPTOOL, the AXOID for AXDB and the temporary AXOID if the object is recovered by the crawler.

For each AXObject a set of extra info can be optionally reported. Each extra info tag is comprised of a couple of field and value. In this case also, when the AXDB schema will be stable we will be able to limit the valid content for field tag.

In addition an error field tag with a list of fields that have not allowed the resolution of the query have been added. errorfield contains an attribute that allows to identify which subsystem has failed the query.

15.3.4.1 XML results example file

In this section are collected sample XML file compliant to the proposed schema that can be useful to understand the practical structure of the results of a query.

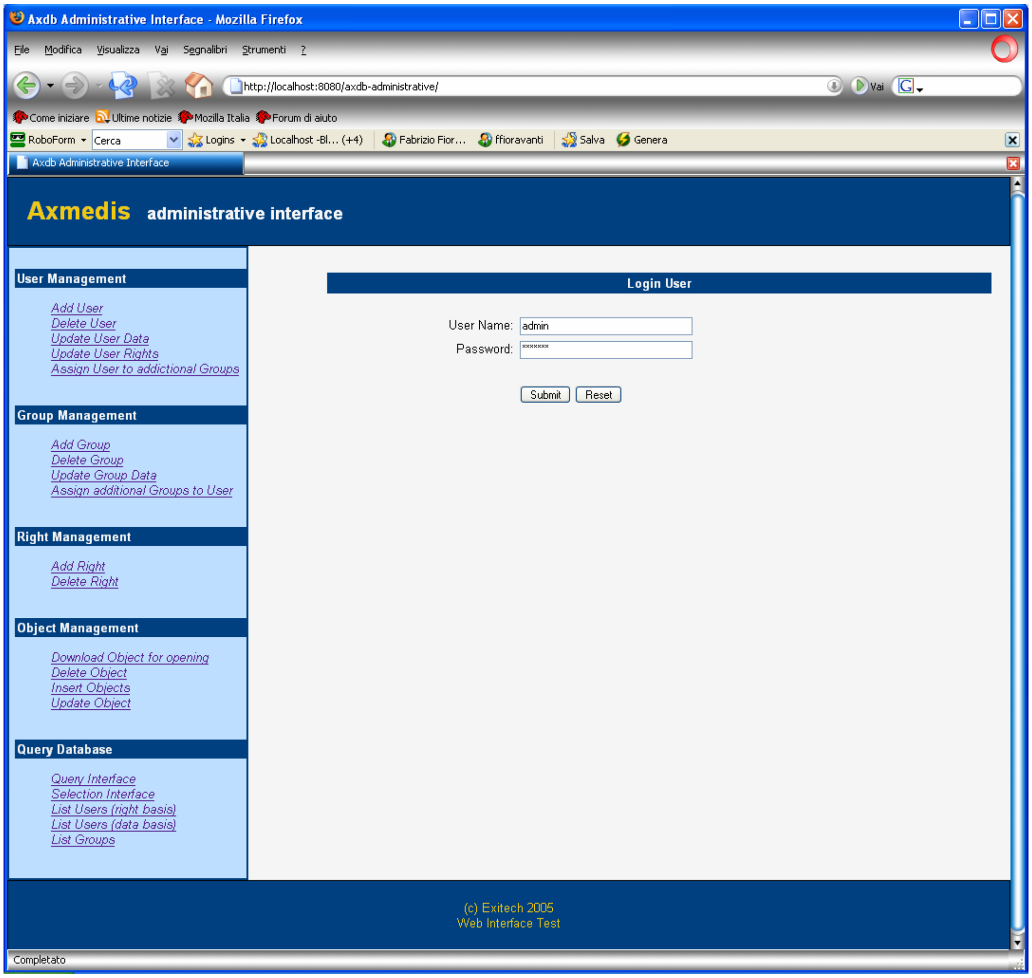
```
<?xml version="1.0" encoding="UTF-8"?>
<queryresults xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="AXQS-Result-v-1-
3.xsd">
    <AXObject>
        <AXEPTOOL>
            <peers>
                <peerID>12234</peerID>
```

```
                                <peerID>56454</peerID>
                                <peerID>12784</peerID>
                        </peers>
                        <AXOID>12248766-gftr</AXOID>
                </AXEPTOOL>
                <extrainfo>
                        <field>Author</field>
                        <value>Ramazzotti</value>
                </extrainfo>
                <extrainfo>
                        <field>YearOfPublication</field>
                        <value>1996</value>
                </extrainfo>
                <extrainfo>
                        <field>Duration</field>
                        <value>315</value>
                </extrainfo>
        </AXObject>
        <AXObject>
                <CRAWLER>
                        <TAXOID>5383999-temp</TAXOID>
                </CRAWLER>
                <extrainfo>
                        <field>Author</field>
                        <value>Ramazzotti</value>
                </extrainfo>
                <extrainfo>
                        <field>YearOfPublication</field>
                        <value>1997</value>
                </extrainfo>
                <extrainfo>
                        <field>Duration</field>
                        <value>417</value>
                </extrainfo>
        </AXObject>
        <AXObject>
                <AXDB>
                        <AXOID>35423912843.fkhwgywe</AXOID>
                </AXDB>
                <extrainfo>
                        <field>Author</field>
                        <value>Ramazzotti</value>
                </extrainfo>
                <extrainfo>
                        <field>YearOfPublication</field>
                        <value>1995</value>
                </extrainfo>
                <extrainfo>
                        <field>Duration</field>
                        <value>234</value>
                </extrainfo>
        </AXObject>
</queryresults>
```

Here follows another example of results considering that crawler will report an error

```
<?xml version="1.0" encoding="UTF-8"?>
<queryresults xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="AXQS-Result-v-1-
3.xsd">
<errorfields source="CRAWLER">
        <fieldname>/descriptor/mine/duration</fieldname>
        <fieldname>/descriptor/dcmi/authir</fieldname>
</errorfields>
        <AXObject>
        <AXEPTOOL>
            <peers>
                <peerID>12234</peerID>
                <peerID>56454</peerID>
                <peerID>12784</peerID>
            </peers>
            <AXOID>12248766-gftr</AXOID>
        </AXEPTOOL>
        <extrainfo>
            <field>Author</field>
            <value>Ramazzotti</value>
        </extrainfo>
        <extrainfo>
            <field>YearOfPublication</field>
            <value>1996</value>
        </extrainfo>
        <extrainfo>
            <field>Duration</field>
            <value>315</value>
        </extrainfo>
    </AXObject>
    <AXObject>
        <AXDB>
            <AXOID>35423912843.fkhwgywe</AXOID>
        </AXDB>
        <extrainfo>
            <field>Author</field>
            <value>Ramazzotti</value>
        </extrainfo>
        <extrainfo>
            <field>YearOfPublication</field>
            <value>1995</value>
        </extrainfo>
        <extrainfo>
            <field>Duration</field>
            <value>234</value>
        </extrainfo>
    </AXObject>
</queryresults>
```

These functionalities have been partially used in the AXMEDIS database administrative interface. They have been implemented during this period.



User Name: admin

Password: *******

(c) Exitech 2005
Web Interface Test

AXMEDIS Administrative Web Database Interface provides to the AXMEDIS administrator access to the main functionalities of AXMEDIS that are related to the database.
Among the planned functionalities, the implemented functionalities also in form of a working prototype without frills, are:

- Add user
- Delete User
- Update user data
- Update user rights
- Add Group
- Delete Group
- Update group data
- Assign users to group
- Assign groups to user
- Lists user of the basis of the rights
- List users on the basis of the user data
- List group

Regarding the work performed on query support a large work have been done regarding the query adaptor for AXDB, that is the module that manages the query on the AXDB and translates the query from XML to XML managing also multilingual problems and optional fields related problems.
The current specification can be found in the specification document DE 3.1.2E Database-Gathering version 2.3 or higher.
Web service are described in the specification document that have been realized are the following:

- AXDBQuerySupport (only sync functionalities have been implemented according to what has been scheduled)
- QuerySupport Web service interface (only sync functionalities have been implemented according to what has been scheduled)

Moreover the work that has been done is not only the implementation of the functionalities but also the test of all the functionalities also implementing other web service (such as Crawling web service, Licence web service, P2P webservice) in order to simulate distribution and collection of queries.

16 Collector Engine and Query Support in the Database of Crawler Results Integrated Database

16.1 Collector Engine

The *Collector Engine* will allow migrating content from the *Crawled Results Integrated Database* to the AXMEDIS format and database.
Using the focuseek Watch Manager the user can make a persistent query to select the content to be acquired (e.g. all the songs of Ramazzotti) and when a new content satisfying the query is present or it is updated the acquisition/update of the object is performed.
The acquisition/update is done using a *javascript* script to:

1. access to the metadata associated with the "document" to be acquired stored in the Crawler DB
2. create an AXMEDIS object integrating the content and the metadata acquired
3. use the fingerprinting tools to add new descriptors to the content
4. put the object in the AXMEDIS Database

The *Collector Engine* is implemented using the *AXMEDIS Content Processing Rule Engine* (see DE 4.3.1) which is used to execute JS scripts for content & metadata acquisition from the *Crawled Results Integrated Database.* To allow the interaction of javascript rules with searchbox tools specific JS classes have been realized.

For example the following script makes a query using term "string" and for each document displays the mimetype, the document size and all the metadata associated with the document.

```
// connects with searchbox
var sb = new AXSearchbox();

sb.Host = "localhost";
sb.Port = "2200";
sb.Username = "admin";
sb.Password = "password";

// creates the QuerySpec object with the query to be performed
var qs = new QuerySpec();
qs.Collection = -1; // search on all the archives
qs.Parser = QueryParser.ALGPARSER;
qs.Info = QueryInfo.INFO_CONTEXT;
qs.View = QueryView.VIEW_PUBLISHED;
qs.Sort = QuerySort.SORT_STANDARD;
qs.QueryString = "string"; // the query string
qs.FirstDoc = 0;
qs.LastDoc = 1;

// makes the query and gets the results in qr array
var qr = new Array();
var maxres = sb.Query(qs, qr);

// i
var i, j;
for(i = 0; i < qr.length; ++i)
{
      print(qr[i].ID+" "+qr[i].Url);

      // gets the document by ID from the searchbox document store
      var doc = sb.GetDocument(qr[i].ID);
      print("-> "+doc.MimeType+" ["+doc.Size+"]");
      // extracts the document to the file system
      doc.Write("C:/demo/file."+extFromMimeType(doc.MimeType));

      // gets the metadata of document by ID
      var meta = sb.GetDocumentMetadata(qr[i].ID);
      for(j = 0; j < meta.length; ++j)
      {
            print("--> "+meta[j].Key+"["+meta[j].Slice+"]="+meta[j].Value);
      }
}
```

16.1.1 Work Performed

1. JS class AXSearchbox as main wrapper for SB SOAP methods, with endpoint defaults retrieved from the registry and properties to change them at runtime.

- JS bindings for the following searchbox SOAP methods, implemented as methods of the AXSearchbox class:
  - Query
  - AddDocument
  - RemoveDocument
  - GetDocument
  - GetDocumentURL
  - ApplyMetadataTemplate
  - DeapplyMetadataTemplate
  - EnumAppliedMetadataTemplates
  - NormalizeURL.

2. Related JS wrapper class for all the SOAP data structures involved with the previous methods.
3. Opaque JS class for handling of binary data, with Load/Save methods for FS import/export.
4. JS method GetDocumentMetadata to retrieve metadata associated with a document, implemented as method of the AXSearchbox class.

16.2 Query Support

The Query Support in the Crawler Database has to adapt AXMEDIS queries to focuseek queries

16.2.1 Work Performed

1. Queries in AXMEDIS syntax are translated to searchbox internal queries and then searchbox query results are translated back into AXMEDIS format. Full AXMEDIS syntax, as specified in "Framework and Tools Specifications (Database and Gathering)" and subsequent modifications, is

supported. Two unit test suites (one testing the query translation, the other one testing the results translation) and a standalone test program are delivered together with the C++ translator classes.

2. A C++ web service has been added to focuseek searchbox implementing the Query_Support web service described in the aforementioned specification document. The service uses the translators described in (1).

Limitations and known problems:

1. Returning the errorfields tag in the query results can be inefficient when a large number of AXInfo filed tags are specified in the query.
2. Due to performance considerations and searchbox intrinsic limits REGULAR EXPRESSION similarities are not implemented.

17 Query Support for PC using Web Interface

17.1 User/Web Interface

The demonstrator web interface provides the possibility to query the AXDB-Server for information about the available multi-media content. The AXDB server provides a powerful query language that is too complex to be handled by an end user in this case. For this purpose the AXMEDIS specification defines the simplified query language schema. The user/web interface therefore creates a request based on this language schema, which is processed by the Query Support for Distribution Channels module as described in the next section.

After the server has evaluated the user request and returned the results, the demo client presents a list of matching content to the user. The user then chooses the content she/he wants to see or goes back to enter another query. If the user requests content, this request is delegated to the Programme and Publication Tools module, which prepares the content, based on the determined AXOID.

Since the AXMEDIS platform needs to support also various other devices besides personnel computers, it is necessary that the Web interface contains further functionalities. Besides the base functionality described above, the provision and determination of Distribution and Client Profiles is needed. On the one hand these profiles are used to refine the queries of a user and on the other hand to generate the requested content adapted for the specific device.

17.1.1 State of the Art

The user/Web Interface is based on the Apache Struts Framework. For the processing of the Client Profiles, which describe the specific properties of a user device in more detail, the User/Web interface uses the CC/PP protocol. The CC/PP protocol is the result of the CC/PP Work Group, which is now continued by the DEVICE Independence Work Group (DI WG) of the W3C, and permits the definition and the exchange of device dependent attributes. The web interface uses the User Agent Profiles (UAProf), which is the first large-scale deployment of CC/PP.

17.1.2 Problems

Web based input of the user query

The primary task of this module is the conversion of the query information collected by means of an HTML form to a well-defined XML data structure (Simple Query Language). In order to reduce the number of client/server requests and to improve the user friendliness, the input of the user query was optimized using JavaScript, which however requires a server-side work necessary afterwards because of its dynamic nature.

Support for different devices

The structure and the vocabularies of the CC/PP has become a W3C recommendation, 15 January 2004. Unfortunately, currently, no standardized exchange protocol is defined. There are two main ways of transporting CC/PP over HTTP: Using the CC/PP Exchange Protocol defined by the DI Workgroup, or using the UAProf W-HTTP Protocol defined by the WAP-Forum (Open Mobile Alliance). Since not all devices support the CC/PP protocol, there is still a need for a fallback solution.

Collaboration with other AXMEDIS modules

The query and the profile retrieved had to be communicated to the AXDB server via the Distribution Channels Module. This module can be integrated as a Java library which wraps the logic required for the web service communication. The Programme and Publication Tool on the other hand had to be accessed on a stream based level like a socket connection which is passed thru to the HTML response of the client. A requested movie file e.g. can not be completely loaded and then forwarded to the user. This procedure will lead to a not acceptable latency.

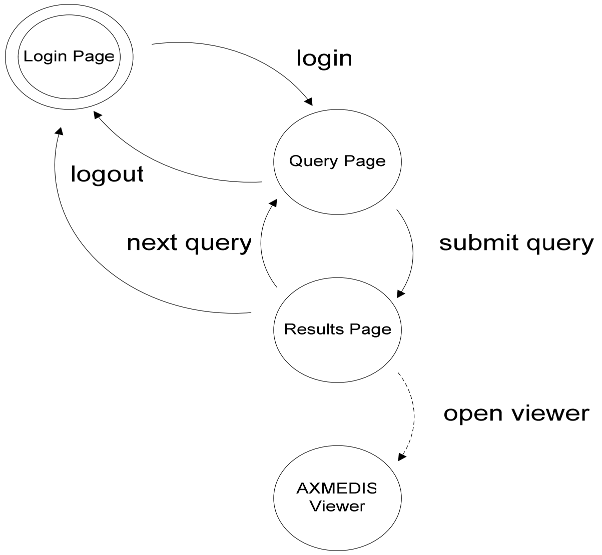
17.1.3 Work performed

The current implementation permits the Web based input of search parameters, the preparation and conversion of these request parameters into the simplified query format, to call the responsible AXMEDIS component for subsequent treatment of the query and the presentation of the query results as list. The handling of the Client and Distribution Profiles, as well as the communication between Web interface and P&P Tool are not yet completely implemented. The latter requires further coordination with the partners involved.

In the following the different steps are described in more detail.

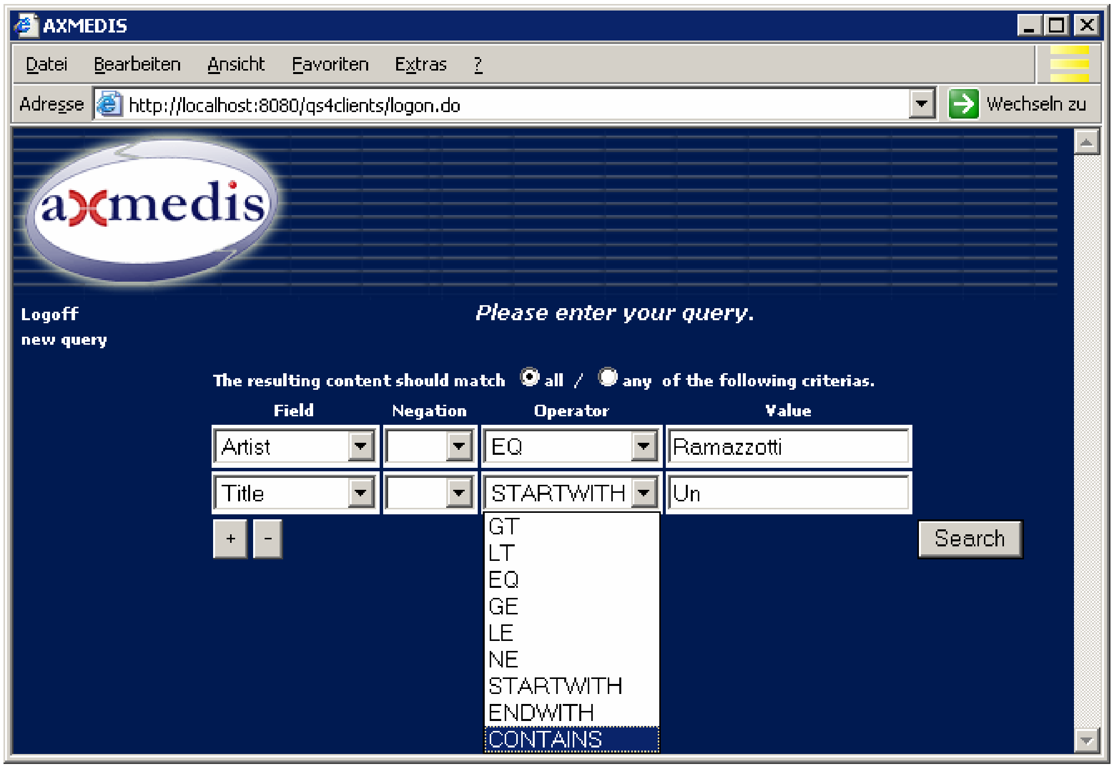
Interaction of the system with the user

The following state chart shows the interaction between the users and the AXMEDIS framework. The implementation of the user/Web interface is based on the Struts Framework of Apache. More information about this framework can be found at [http://struts.apache.org/index.html].



State chart of web application

After the user profile is determined, the user is requested to enter her/his query. This procedure is supported on the client with JavaScript to minimize the number of client/server requests and therefore to ensure a smooth and intuitive operation of the query interface.

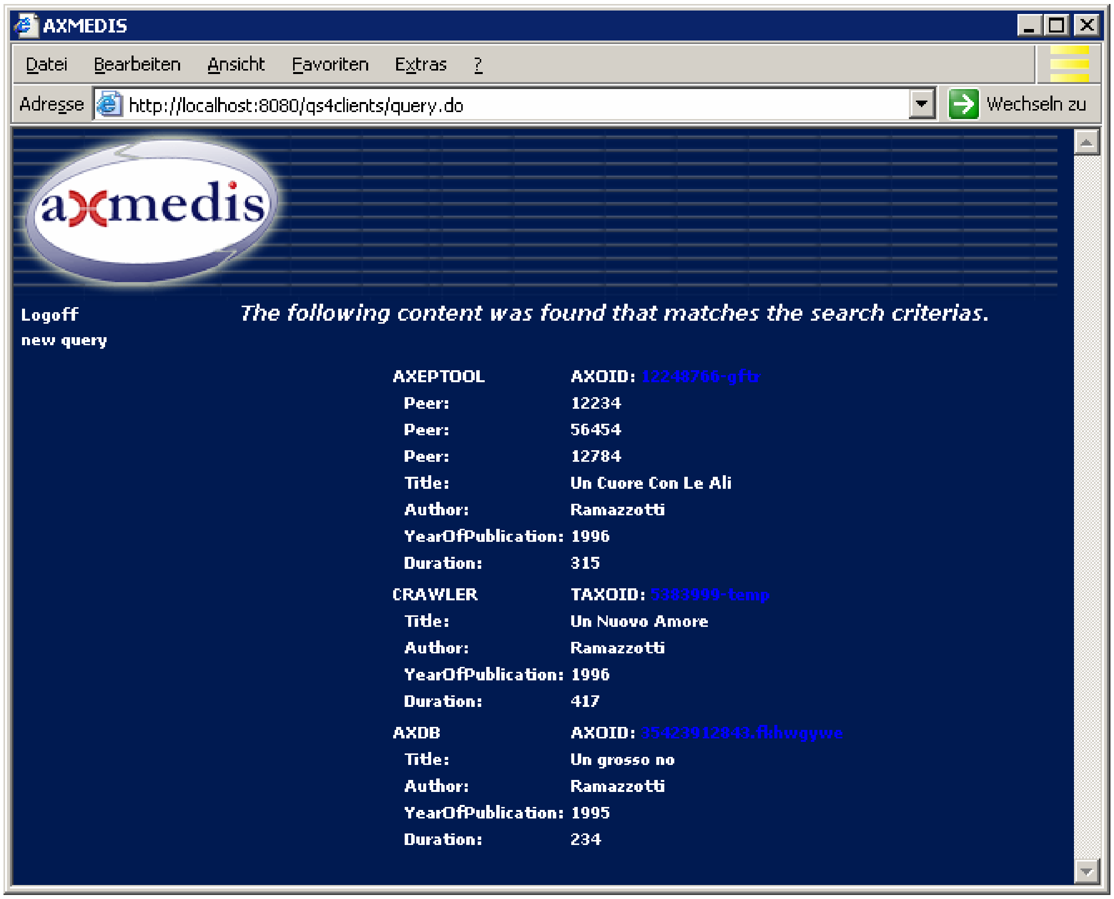


Please enter your query.

The resulting content should match ● all / ○ any of the following criterias.

Field	Negation	Operator	Value
Artist		EQ	Ramazzotti
Title		STARTWITH	Un

After the user query was sent to the distribution server, the results that were determined by the AXMEDIS system are presented to the user.



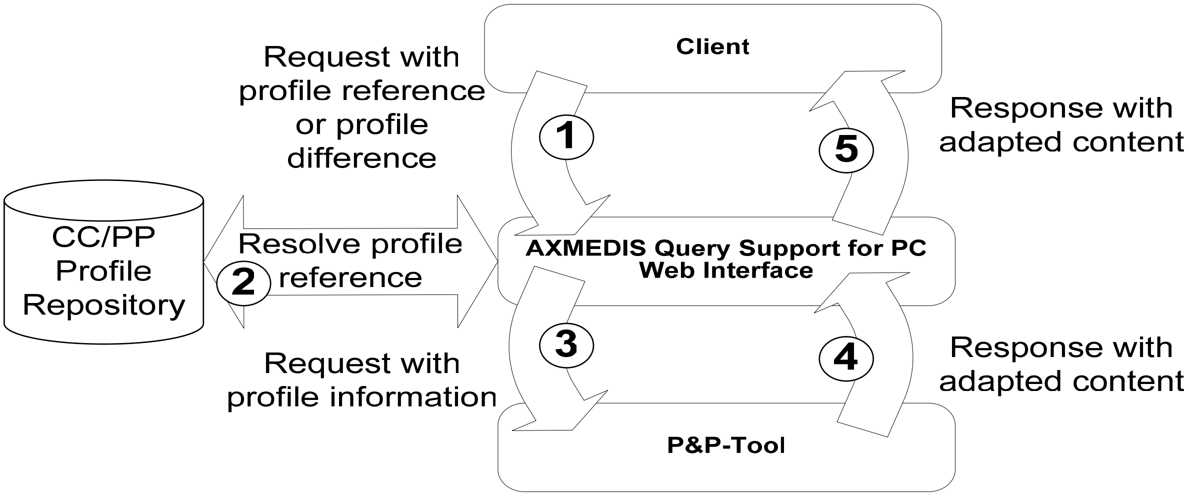
The following content was found that matches the search criterias.

AXEPTOOL	AXOID: 12248766-gftr
Peer:	12234
Peer:	56454
Peer:	12784
Title:	Un Cuore Con Le Ali
Author:	Ramazzotti
YearOfPublication:	1996
Duration:	315
CRAWLER	TAXOID: 5383999-temp
Title:	Un Nuovo Amore
Author:	Ramazzotti
YearOfPublication:	1996
Duration:	417
AXDB	AXOID: 35423912843.fkhwgywe
Title:	Un grosso no
Author:	Ramazzotti
YearOfPublication:	1995
Duration:	234

Why Struts?
The Struts Framework is a slim and efficient Open Source Framework for thin client Web applications that shines by its prevalence and the good support with tools. The framework achieves this by successfully implementing the Model View Controller paradigm (struts.apache.org)

Besides Struts also the Cocoon Framework was taken into account. This Framework offers a better support for CC/PP (via DELI) as well as a better expandability concerning various devices. The larger complexity and thereby increased implementation cost of the framework made this framework however unsuitable for the implementation of the client demonstrator (cocoon.apache.org)

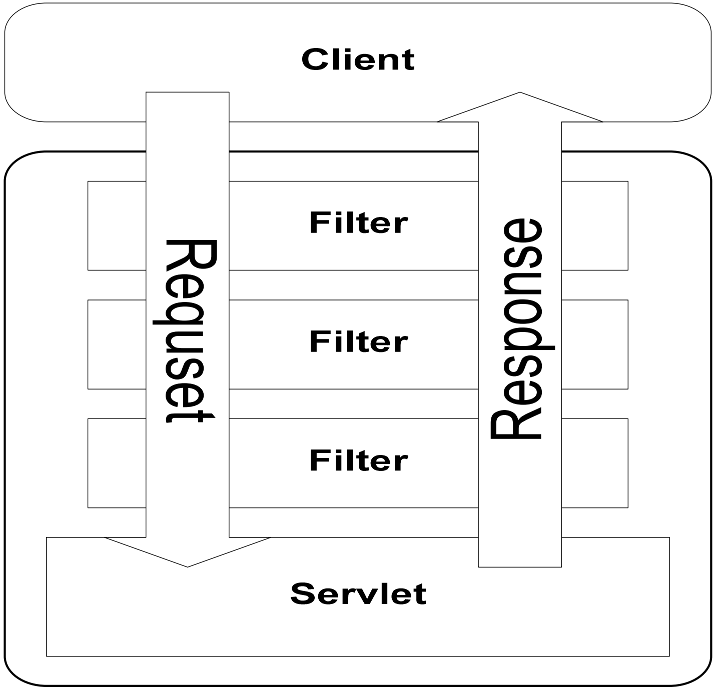
Client Profile Resolution
The user/Web interface retrieves additional information with each query of a CC/PP or UAProf capable device in the HTML headers. The Web interface can use this information to create a Client Profile.
The resulting profile is used in two different ways. On the one hand the query to the AXDB server can be refined and adjusted, on the other hand the profile is necessary for the Programme and Publication Tool to prepare and adapt the requested content for the device using the profile information.


Client
Request with profile reference or profile difference
1
5
Response with adapted content
CC/PP Profile Repository
Resolve profile reference
2
AXMEDIS Query Support for PC Web Interface
3
4
Request with profile information
Response with adapted content
P&P-Tool

Client Profile Processing

Both available transportation protocols, HTTP-Ex as well as W-HTTP, are structured in a similar way and can be treated equally up to a certain level. However, the UAProf protocol offers a better selection of vocabularies, and therefore it was chosen for the implementation. If the device does not provide any profile information, the user/Web interface assumes that the device is a standard PC with typical characteristics and falls back to a default profile that is stored on the distribution server.

The determination of the Client Profiles is independent of the actual query and is done using a servlet filter mechanism. Servlet filters are powerful tools that are available to web application developers using the Servlet specification version 2.3 or higher. Filters are designed to be able to manipulate a request or response (or both) that is sent to a web application, yet provide this functionality in a method that will not affect servlets and Java Server Pages that are used by the web application unless that effect is the desired. A good way to think of servlet filters is as a chain of steps that a request and response must go through before reaching a servlet, JSP, or static resource such as an HTML page in a web application." (http://www.onjava.com/pub/a/onjava/2003/11/19/filters.html)



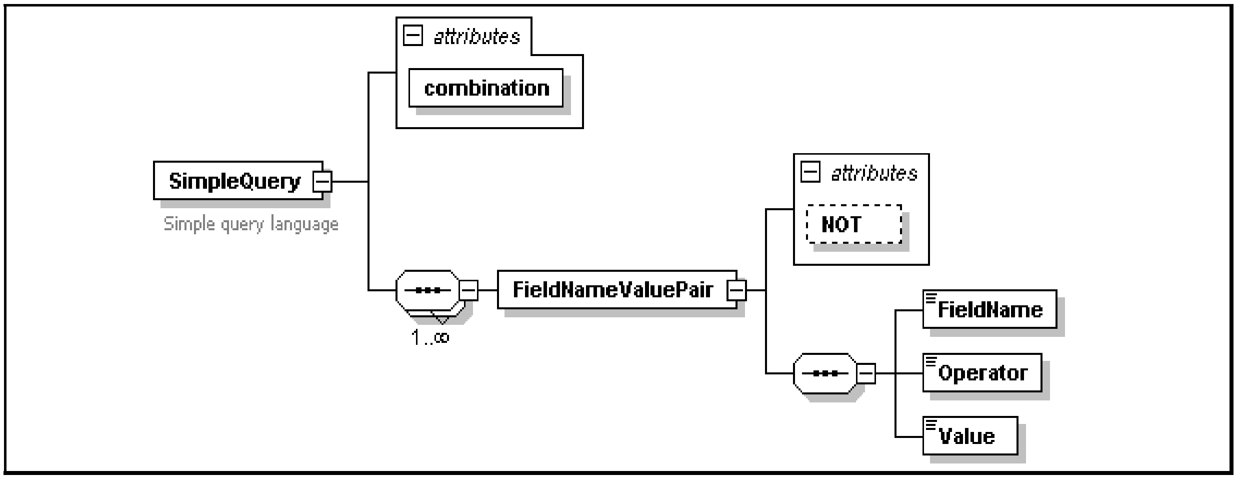
Communication with the AXDB Server

This section describes the different processing steps between the submission of the query and the presentation of the search results.

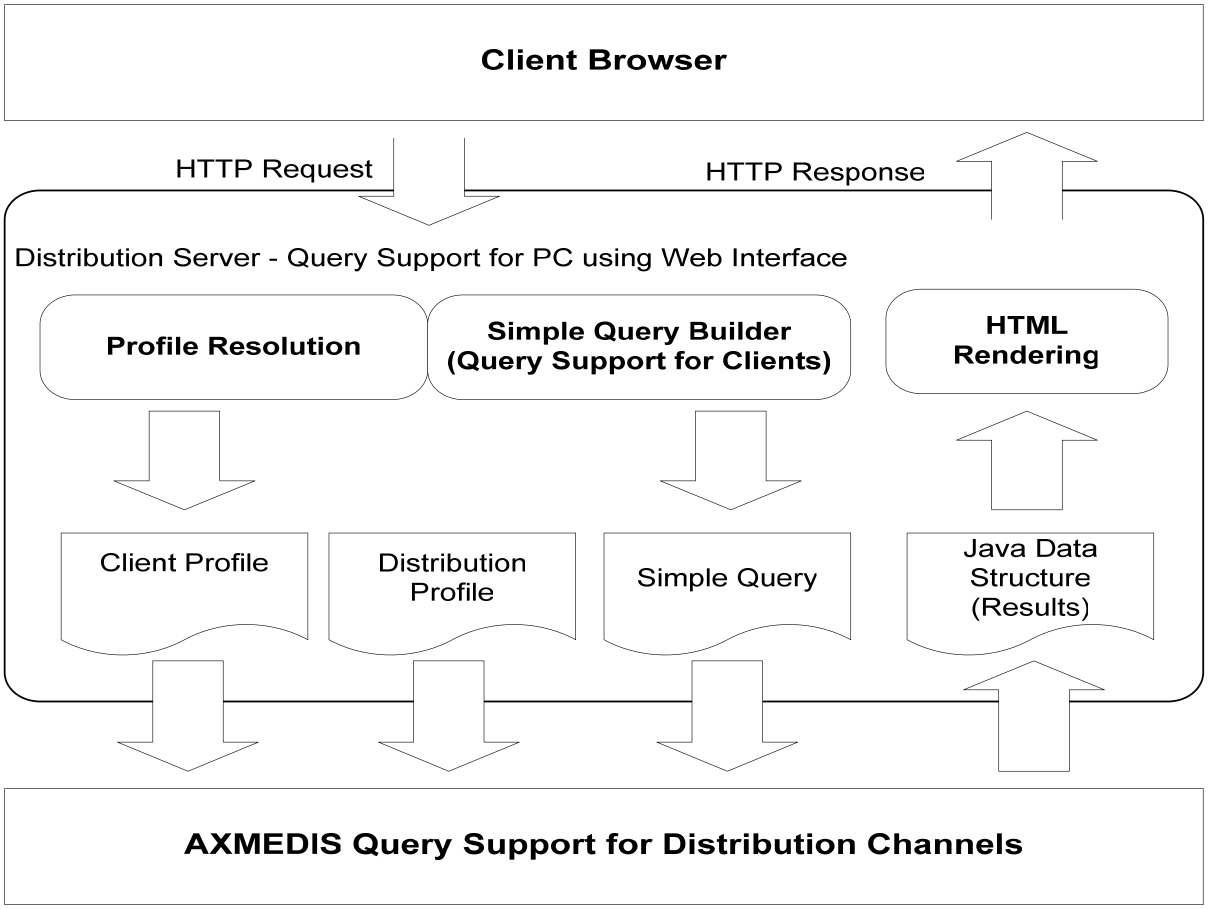
After the Client Profile is determined the request parameters are sorted and grouped into the respective query terms. By means of the JAXB (Java XML being thing) API they are then transferred into XML. This is based on the XML schema shown below.

The schema is an extension of the pattern described in the specification document. The new pattern permits the specification of a logic connection of all search terms, as well as the specification of comparison operators concerning the individual terms. Furthermore the pattern defines a range of valid values for the individual elements, which permits a simple validation of the query at an early stage. A graphical representation of the schema can be found below the following text.

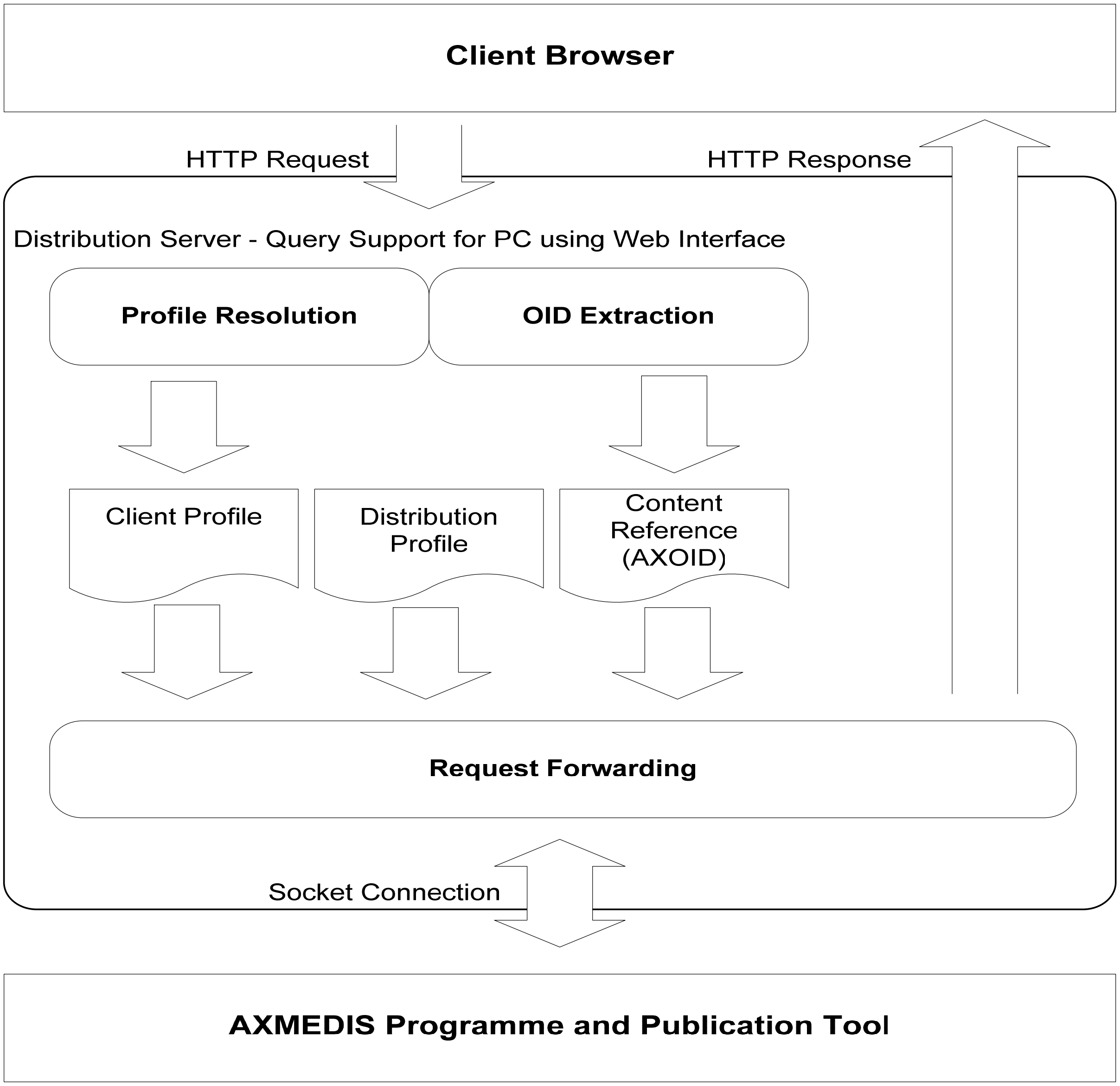
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="SimpleQuery">
        <xs:annotation>
            <xs:documentation>Simple query language</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
                <xs:element name="FieldNameValuePair">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="FieldName">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:enumeration value="Title"/>
                                        <xs:enumeration value="Artist"/>
                                        <xs:enumeration value="Subject"/>
                                        <xs:enumeration value="Keywords"/>
                                        <xs:enumeration value="Description"/>
                                        <xs:enumeration value="Media Type"/>
                                        <xs:enumeration value="Year"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="Operator">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:enumeration value="GT"/>
                                        <xs:enumeration value="LT"/>
                                        <xs:enumeration value="EQ"/>
                                        <xs:enumeration value="GE"/>
                                        <xs:enumeration value="LE"/>
                                        <xs:enumeration value="NE"/>
                                        <xs:enumeration value="STARTWITH"/>
                                        <xs:enumeration value="ENDWITH"/>
                                        <xs:enumeration value="CONTAINS"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="Value" type="xs:string"/>
                        </xs:sequence>
                        <xs:attribute name="NOT" type="xs:boolean" use="optional" default="false"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="combination" type="andOr" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:simpleType name="andOr">
        <xs:restriction base="xs:string">
            <xs:enumeration value="AND"/>
            <xs:enumeration value="OR"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```



The resulting query represented as a XML document is then passed on to the module "Query Support for Distribution Channels" together with the Distribution and Client Profiles. After processing the query this module returns a Java data structure, which contains information of the determined contents. This data structure is then transferred into an appropriate HTML representation and sent back to the client. The following chart shows the operational sequence described.


Client Browser
HTTP Request
HTTP Response
Distribution Server - Query Support for PC using Web Interface
Profile Resolution
Simple Query Builder
(Query Support for Clients)
HTML
Rendering
Client Profile
Distribution
Profile
Simple Query
Java Data
Structure
(Results)
AXMEDIS Query Support for Distribution Channels

Communication with the P&P Tool
This section describes the processing steps between the selection of a specific content from the presented list of results and the retrieval of this content from the P&P tool. After the determination of the Client and the Distribution Profile and the desired content (by its AXOID), the resulting request is sent to the Programme and Publication Tool. The response of the P&P Tools is transferred, if no errors occurred, into a HTTP response. After receiving this HTTP message, the client browser opens an appropriate viewer based on the returned MIME type.



17.2 Query Transcoding and Executing

This module serves as a link between the different distribution channels and the AXDB server. The reduced scope of the Simplified Query Language for a simple interaction with the final user requires an appropriate transformation of this query language into the more complex, but thereby also more powerful, server-side AXDB query language.

Apart from this transformation it is the task of this module to include further information from the Client and Distribution Profile into the query, in order to retrieve suitable content for the client device.

In the next step the refined query is sent via the AXDB web service interface to the AXDB server. The response of the server is again examined, checked for errors and passed on to the querying distribution channel.

17.2.1 Problems

The transformation of the given query to a more complex query structure using a procedural language like Java is a time-consuming process and results in unstable code. That means minor changes in the used schemas can led to many changes at the Java code. The developer has to transfer the XML document into a corresponding Java data structure, navigate inside the source structure while evaluating several attributes, and finally transfers the resulting Java structure back to XML. To refine the user query the necessary information, e.g. resolution of the client device, have to be extracted from the Client Profile. The Java Specification Request (JSR) 188 provides a set of APIs for this purpose, which permits the processing of the Client Profile. However, since all data is available in XML and also the outgoing interface is based on XML, the link of the data from the individual data sources was accomplished using XSL transformations. The use of the APIs offered by JSRs188 would have forced a transformation of the data from XML to Java or the embedding of Java code into the XSL transformation of the existing query. Since these APIs provides no additional value for this processing step it was not used for the implementation.

After the server-side query is generated the module had to invoke the responsible web service provided by the AXDB server to handle this query request. Therefore the module had to implement the required client stub logic and error handling routines.

17.2.2 Work performed

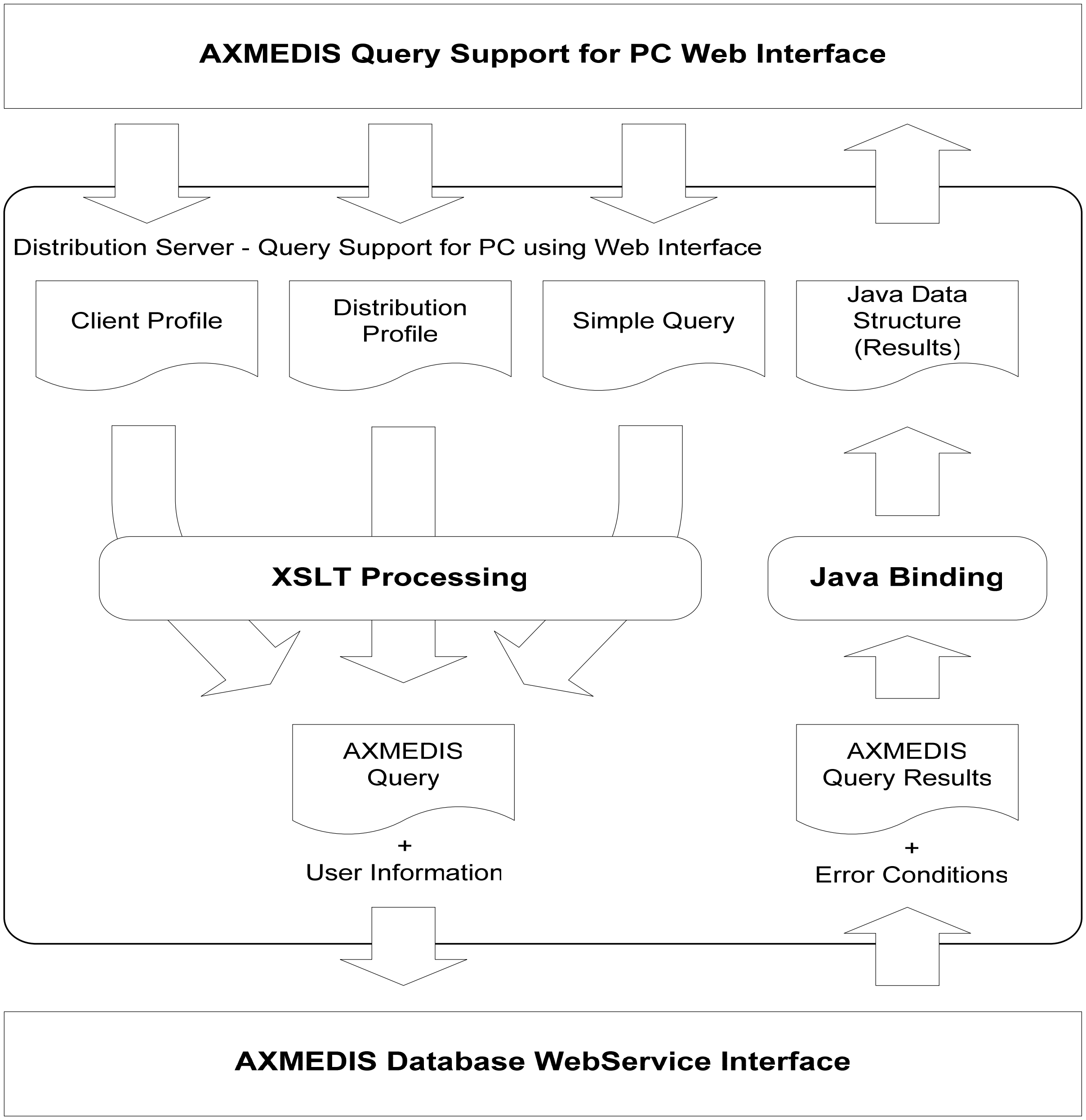
The current implementation transforms the retrieved simple query into the server-side query language, calls the web service provided by the AXDB server, converts the server response into a Java data structure based on the query result schema and passes the result to the requesting user interface. The implementation to invoke the server-side web service is based on the Apache AXIS framework. The server response is converted into the Java data structure using the JAXB (Java XML Binding) API.

17.2.3 Work in Progress

The next step requires the identification of relevant attributes within a client profile. The logic to access these attributes, which are used to refine the user query, had to be embedded into the existing XSL style sheet. Further the XSLT processing has to be modified to pass in the references to the used profiles.

The server response is currently processed based on the AXMEDIS query result schema defined in the specification. Therefore the module had to be adjusted to the query result schema in version 1.2 which provides additional information for error handling.

The following chart shows the operational sequence described.


AXMEDIS Query Support for PC Web Interface
Distribution Server - Query Support for PC using Web Interface
Client Profile
Distribution Profile
Simple Query
Java Data Structure (Results)
XSLT Processing
Java Binding
AXMEDIS Query
+
User Information
AXMEDIS Query Results
+
Error Conditions
AXMEDIS Database WebService Interface

17.3 Production on Demand

17.3.1 State of the Art

Universal multimedia access (UMA) has become the driving concept behind a significant amount of research and standardization activity [Vetro2003]. These concepts are used for terminals of any type to access a rich set of multimedia content for consumption. Ideally this is achieved over dynamic and heterogeneous networks and devices independent of location or time [Vetro2005].

Toward this goal of universal accessibility, techniques for scalable coding and transcoding were developed. An early application of such techniques was in the distribution of television programming from the studio over bandwidth-limited broadcast networks. Shortly after, the challenges in transmitting video over wireless networks and the Internet emerged. While scalable coding and transcoding are quite different approaches, they both offer a means to adapt various coding parameters of the media, such as the bit-rate and spatial/temporal resolution .

An alternative to real-time transcoding and scalable coding is the creation of content in multiple formats and a multiple bit-rates where an appropriate selection is made at delivery time [Smith1999]. While this solution is not intensive for computer processing, it can be highly storage intensive if the received terminals are not limited. In the scenario of no limitations on terminal or access network, Vetro and Timmerer state real-time adaptation is necessary. In this ongoing research, tools are being developed to support universal accessibility and are linked with MPEG-7 [Manjunath2002]. The MPEG-7 framework is based on XML and has standardised a set of description tools and a review of applying tools applied towards the UMA problem is given in [Beek2003]

[MPEG-21] recognised there were missing elements and attempts to fill these elements for multimedia delivery and consumptions across a wide range of networks and devices [Brunet2003]. One aspect of the requirements for MPEG-21 is Digital Item Adaptation (DIA) which is based on a Usage Environment Description (see section 4.7.2 in [MPEG-21]). It proposes the description of capabilities for at least the terminal, network, delivery, user, and natural environment, and notes the desirability of remaining compatible with other recommendations such as CC/PP [W3C2005] and UAProf. In Digital Item Adaptation: Vetro and Timmerer point out that the W3C Consortium [W3C2005] is also engaged in efforts to bridge the gap between multimedia content and devices.

17.3.2 Problems

As discussed earlier, one of the problems of creating content in multiple formats and bit-rates is being storage intensive. Real-time adaptation alone is not a solution when requested objects that have been transcoded as a programme may be required for further distribution.

The solution proposed using the Formatting Engine is to request adaptation when required using the Programme and Publication Engine and store the new object for future distribution. Therefore only original and newly adapted objects are stored. However for on-demand when an object is requested for direct distribution via the PnP Engine, incompatible objects not adapted for the requested distribution channel are required to be adapted in real-time (or as close to real-time as possible) and therefore there is a possible transmission delay. This problem does not apply when objects are processed in programmes delivered in advance of the distribution time using the PnP Editor and subsequently activated from the editor unless immediate distribution is also required.

AXMEDIS Methods for Production On-demand

The AXMEDIS Programme and Publication (PnP) Engine is devised to process programmes of various multimedia content, request adaptation if required, and retrieve the requested objects, either original or newly formatted from the AXMEDIS database for distribution. With respect to on-demand, the request for content is processed immediately when received. Scalable coding and transcoding required for distribution over different terminals and networks, is achieved but utilising the functionalities of the AXMEDIS AXEPTool, developed to achieve these techniques using the MPEG-21 standards.

The first prototype of the Programme and Publication Engine will be used for testing and demonstrating the on-demand publications where adaptation is requested from the XML domain. This method takes the restrictions of networks and terminals into consideration and makes adaptation possible in a flexible manner without the need of the digital object. This is achieved when the profile of the AXMEDIS object is incompatible with the destination profile, the PnP Engine interfaces with the Formatting Engine to provide the appropriate processing on the object and provides the means to access the correctly formatted object for distribution .

This application will be implemented in C++ using wxWindows and Xerces libraries as specified in DE3-1-2. This prototype will be developed for as a win32 application only. The engine will be tested using the XML P&P programmes saved using the first prototype of the P&P Editor using the Programme and Publication libraries (the PnPModel) developed to create new programmes and editing existing programmes (AxPnPRule.lib).

The PnPModel (PnP programme) provides description in three sections. The Header element provides general information such as the Programme Name, Programme ID, version, date of production etc; and production information such as Author, affiliation etc. The Schedule element specifies information such as start date and time, the expiration date and time, and the periodicity. The Definition element provides the rest of the information for distribution i.e. the programme including either the object or a selection of objects, the delivery date and time, duration and where to distribute such as the channel id and terminal id. This provides the minimum amount of information for distribution over different channels of various digital media.

The testing tool will accept a XML programme string from on-demand created utilising the programme libraries. The process required is described in the following section Work Done. Using sockets to send the string, the P&P engine will validate the string. A failed process returns and error message to on demand. A successful process will return the prepared test content to the on-demand.

17.3.3 Work Done

Programme and Publication Model. The PnP Model library can be located from the AXMEDIS repository (see DE5.2.1 for repository lay out) The precompiled libraries are located at Framework/lib/pnpmodel/win32 and named axpnpmodel.lib (release) and axpnpmodel_d.lib (debug). To compile the pnpmodel the project files for .NET VC++ are located at Framework/project/pnpmodel/win32. Compilation of the pnpmodel requires wxWindows and Xerces to be installed and two system variables to be set: WXWIN to specify the location of wxWindows-2.4.2 and XERCES to specify the location of xerces-c_2_6_0 (NB. XERCESCROOT is required for compiling Xerces if not using the precompiled libraries available for download).

For on-demand to create a PnP programme, the following steps are required:

1. Create the PnPRule
```
AxPnPRule* newProgramme = new AxPnPRule( )
```

2. Set required parameters to the rule header
```
newProgramme->setFilename(name);
newProgramme->getHeader()->setRuleName(rule_name);
newProgramme->getHeader()->setSoftwareName(software_name);
newProgramme->getHeader()->setSoftwareVersion(software_version);
```

3. Add a selection to the rule distribution and set the required selection parameters
```
AxRuleSelection* newSelection = new AxRuleSelection();
newSelection->setName(selection_name);
newSelection->setValue(selection_XML_String);
newSelection->updateXML();

AxRuleDistribution* newDistribution = new AxRuleDistribution();
newDistribution->setSelection(newSelection);
newDistribution->setStartDate((wxDateTime)date_to_start_distribution);
newDistribution->setStartTime((wxDateTime)time_to_start_distribution);
newDistribution->setChannelId(channel_for_distribution);
newDistribution->setTerminalId(terminal_for_distribution);



newProgramme->appendDistribution(newDistribution);
```

4. Write programme to XML string and send to the PnPEngine
```
char* XMLString = newRule->writeXMLString()
// code to send the XMLString to the PnP Engine
```

Currently, the name is required to be set before `updateXML` is called and must also be included in the selection XML string, this is to validate the Selection string. The timestamp is set at the current time of the creation of a new `AxRuleSelection`. If the start date or start time is left NULL, the engine will assume to run the programme immediately. The PnP Model is designed to be simple to use. It therefore requires the selection XML to be correctly formatted and error checked before being set to the distribution.

Work in Progress

The PnP Engine is currently being implemented and will utilise the PnP Model to load and parse the XMLString received from on-demand. The PnP Engine is designed to utilise the AXEPTool to format the selected objects from the selection for distribution if formatting if required. At the specified time for distribution, the objects specified in the selection are returned to on-demand.

17.3.4 References

[Beek2003] P. van Beek, J. R. Smith, T. Ebrahimi, T. Suzuki, and J. Askelof, "Metadata- driven multimedia access," *IEEE Signal Process. Mag*., vol. 20, no.2, pp. 40–52, Mar. 2003

[Brunet2003] I. Brunet, R. Van de Walle, K. Hill, J. Bormans, F. Pereira, "MPEG-21: Goals and achievements", *IEEE Multimedia*, Vol. 9, No. 2, October-December 2003, pp. 60-70.

[Manjunath2002] B. S. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG 7: Multimedia Content Description Language.* New York: Wiley, 2002

[MPEG-21] MPEG-21 Requirements v.2 , http://www.chiariglione.org/mpeg, Dec. 2003, Last accessed July 2005.

[Smith1999] J. R. Smith, R. Mohan, and C.-S. Li, "Scalable multimedia delivery for pervasive computing," *Proc. ACM Multimedia*, pp. 131–140, 1999.

[Vetro2003] A. Vetro, C. Christopoulos, and T. Ebrahami, Eds., *IEEE Signal Process Mag. - Special Issue on Universal Multimedia Access*, vol. 20, no. 2, Mar. 2003.

[Vetro2005] A. Vetro, and C. Timmerer, Eds., "*Digital Item Adaptation: Overview of Standardization and Research Activities*" IEEE Transaction on Multimedia, vol. 7, no. 3, Jun. 2005.

[W3C2005] W3C World Wide Web Consortium, http://www.w3.org/ Last accessed, July 2005.

[XercesC2005] XML Apache Project, Xerces C++ Parser, http://xml.apache.org/xerces-c/ Last accessed July 2005.

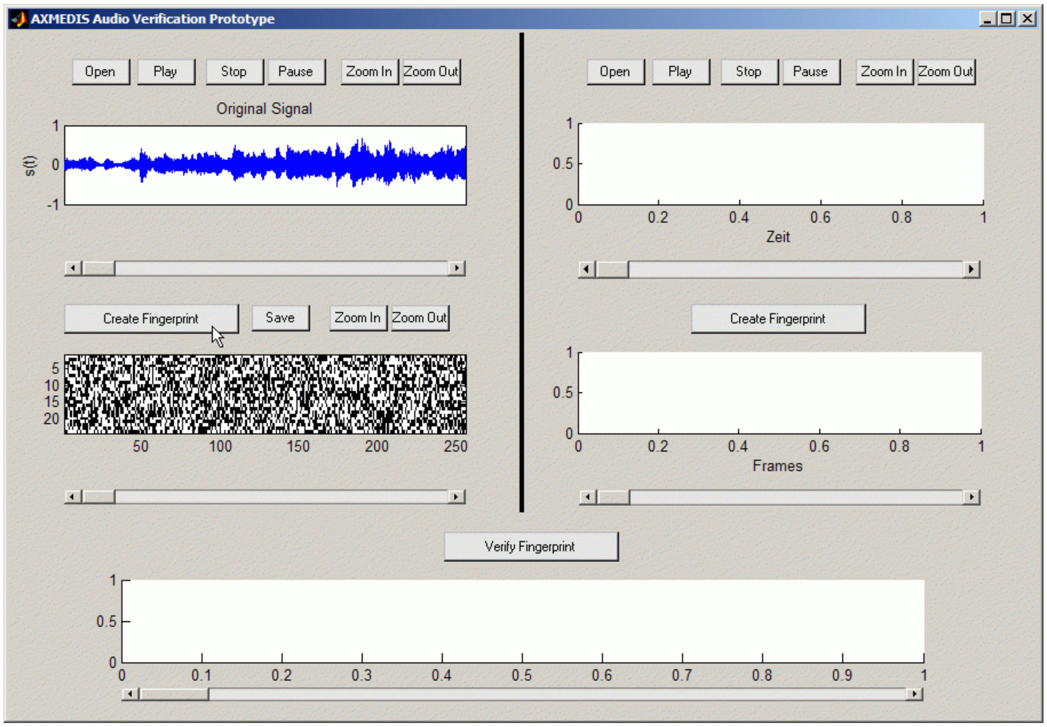
18 Bibliography

The bibliography and the references can be found in the individual sections.

A. Appendix: Fingerprinting Demonstration Prototype

A. Audio Fingerprinting Demonstration

The demonstrator is started in MatLab.

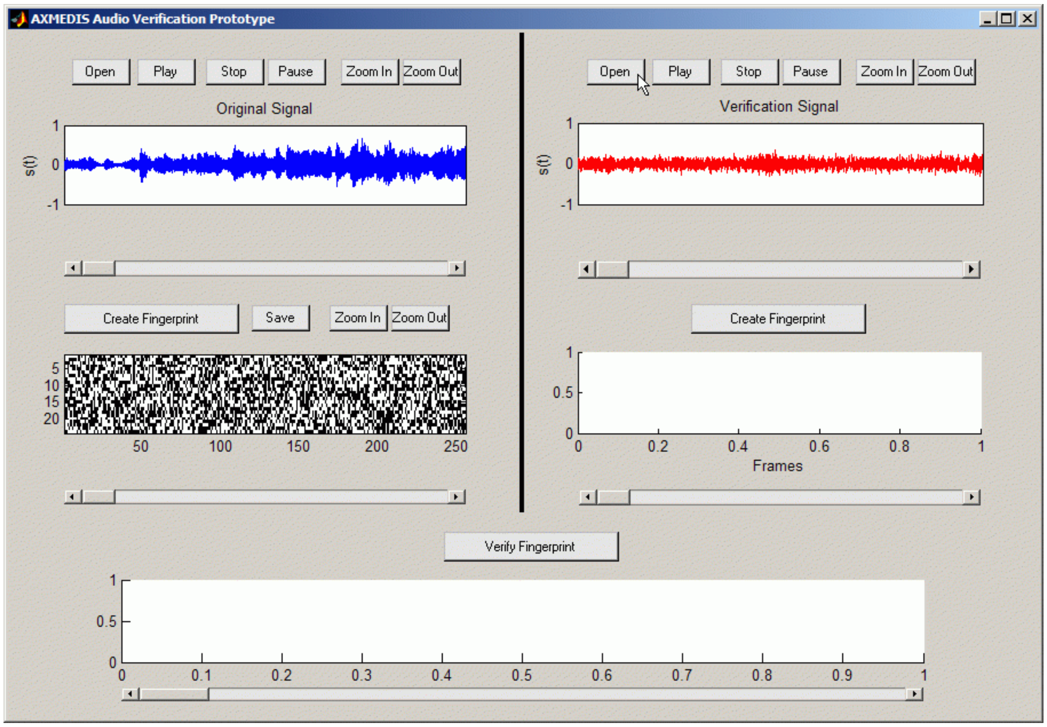


The main window is split in three parts:

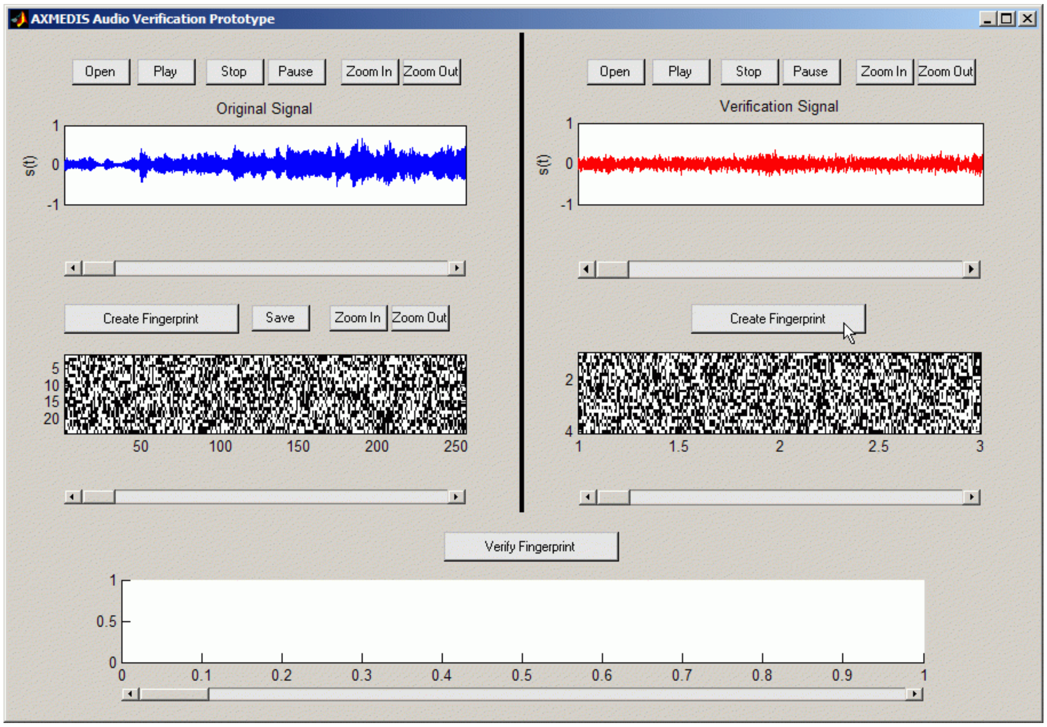
- The (upper) left part of the windows shows the information related to the original signal. The audio can be played and is also visualized as a 1D-temporal function. For this audio signal, the fingerprint can be calculated and viewed.
- The (upper) right part of the window contains the information related to the signal that should be verified. This audio signal is the processed or manipulated. The signal as well as the corresponding fingerprint can be viewed.
- The lower part contains the difference between the fingerprint of the original and the fingerprint of the signal that should be verified.

The demonstration starts with loading the original signal (as shown above). For this signal the fingerprint is calculated.

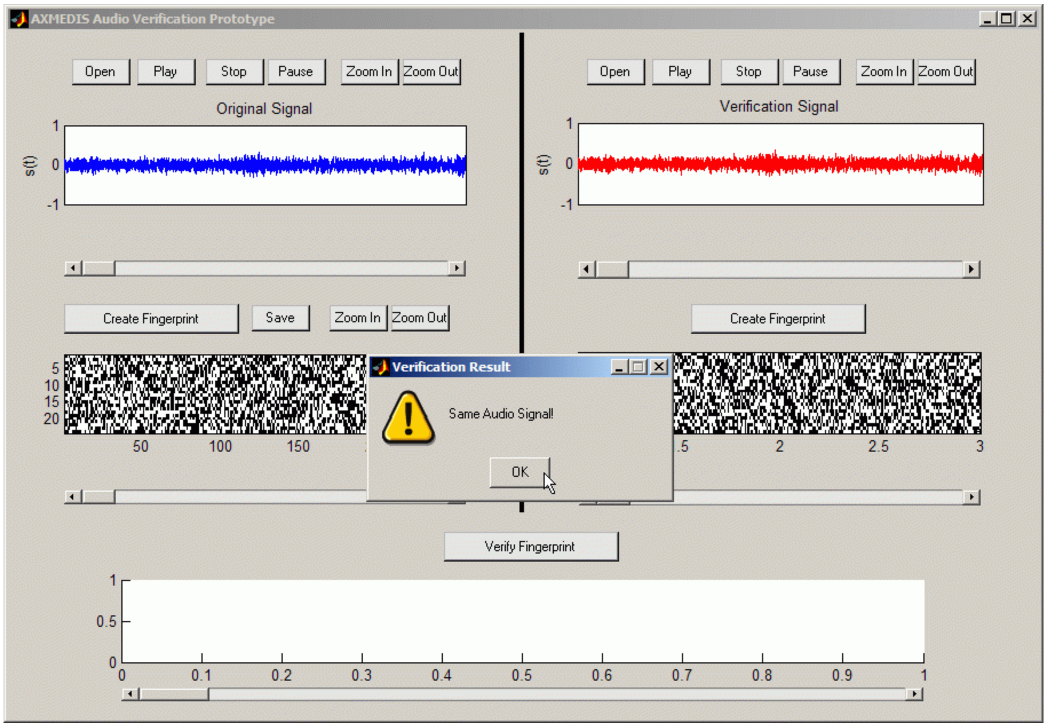
In the second step, the verification signal is loaded as shown below.



After loading the reference signal, its fingerprint is calculated.

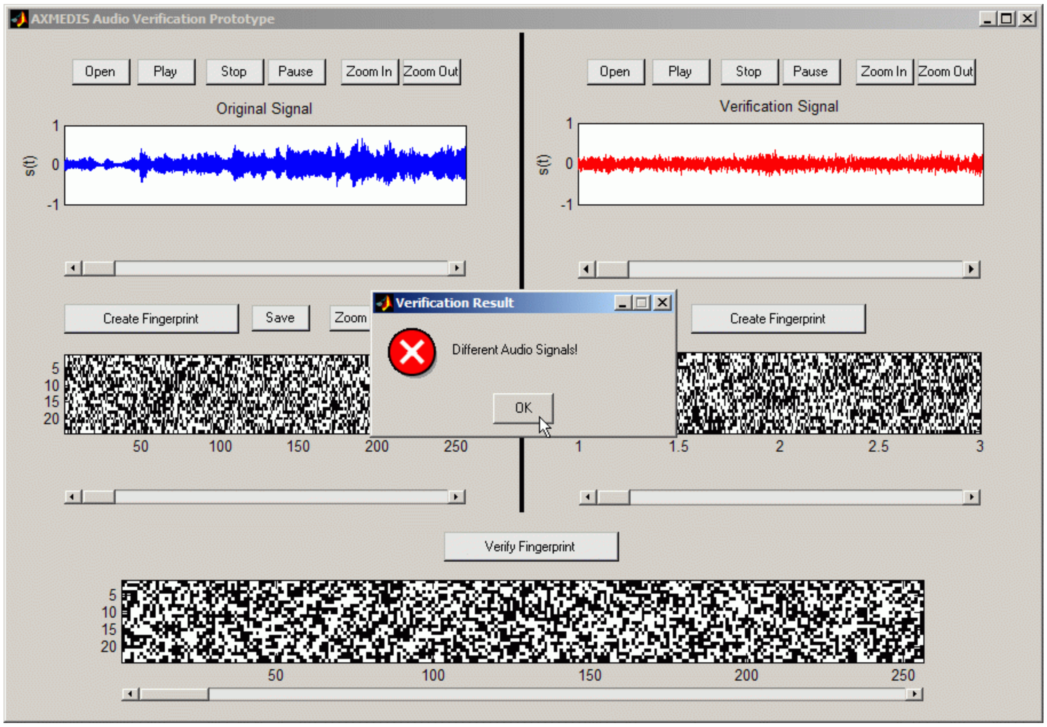


The fingerprint of the original signal and the reference signal are compared. If the signals are similar, a corresponding message box opens:



Same Audio Signal!

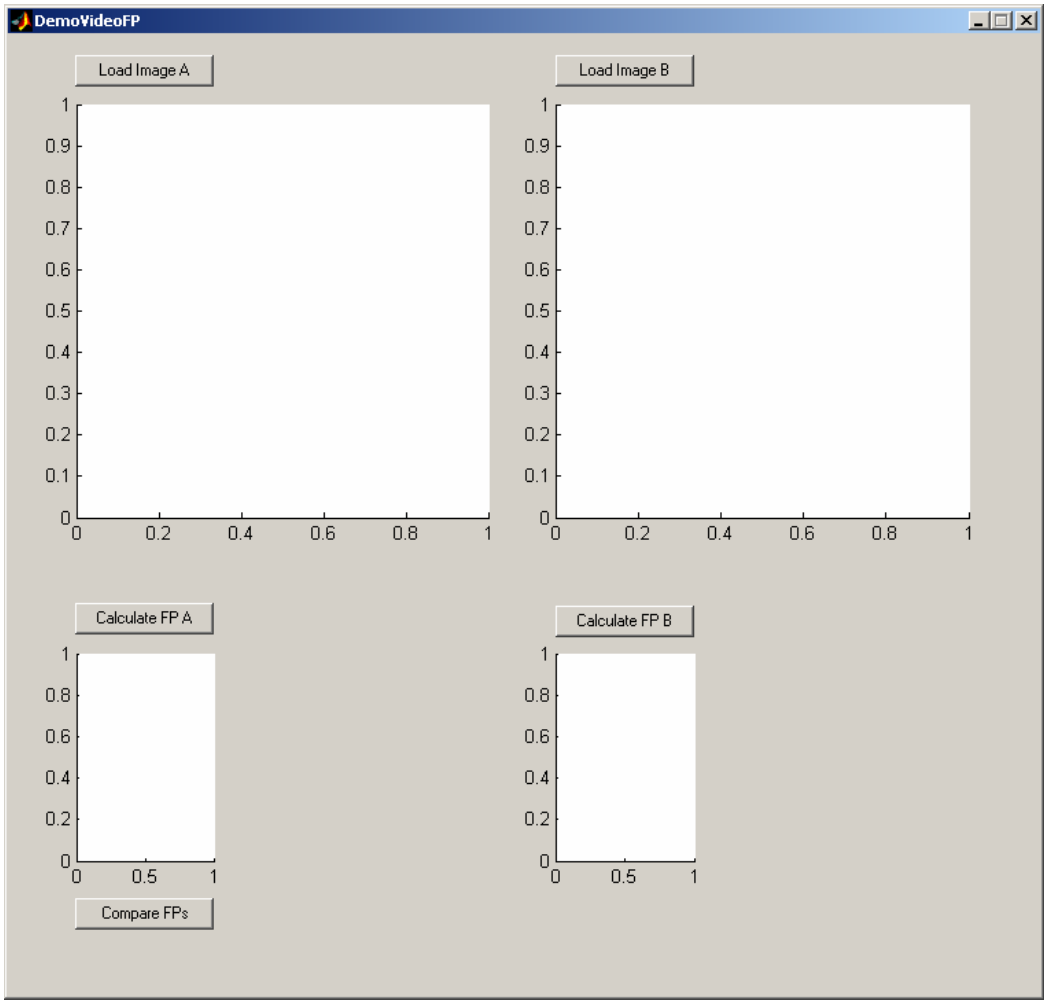
In the case of different signals, an error is shown:



Different Audio Signals!

B. Image Fingerprinting Demonstration

When the application is started in MatLab, a dialog is shown.

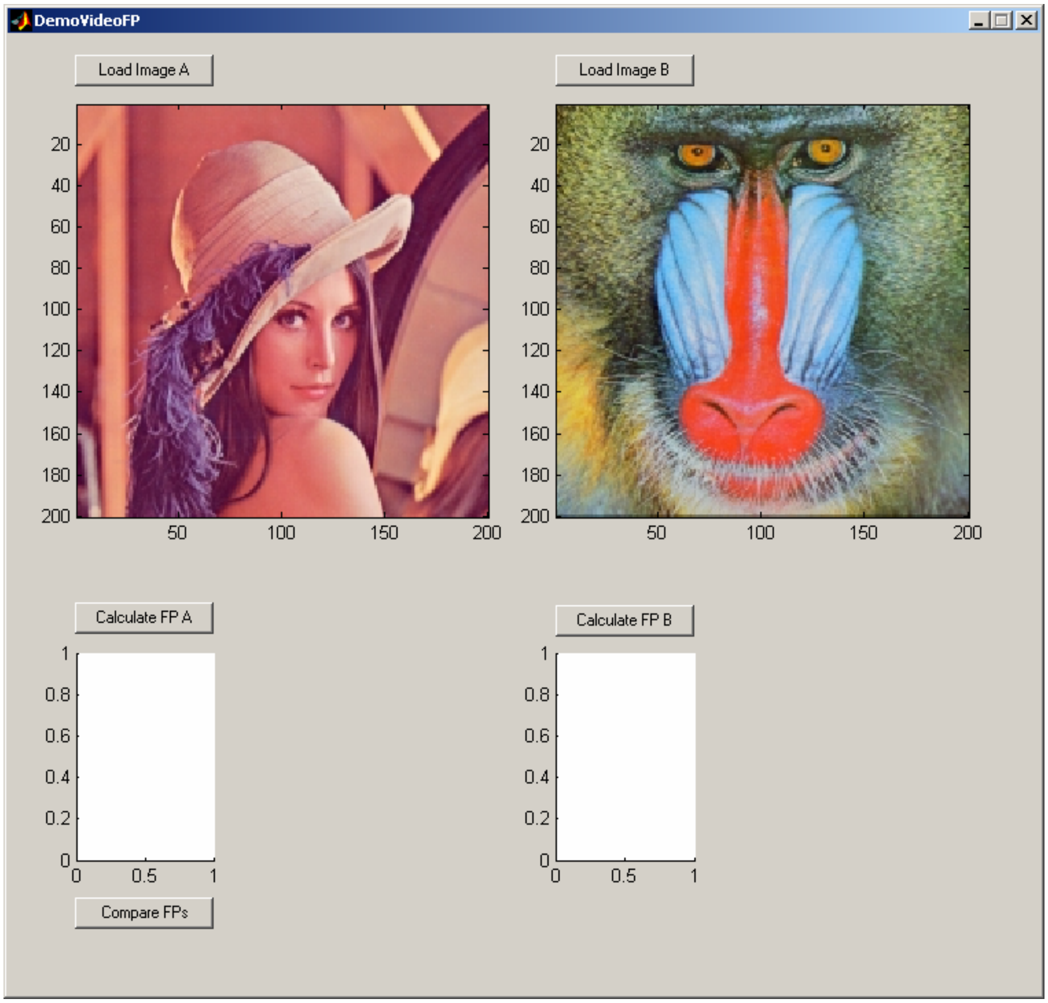


This dialog consists of four buttons, which allows to

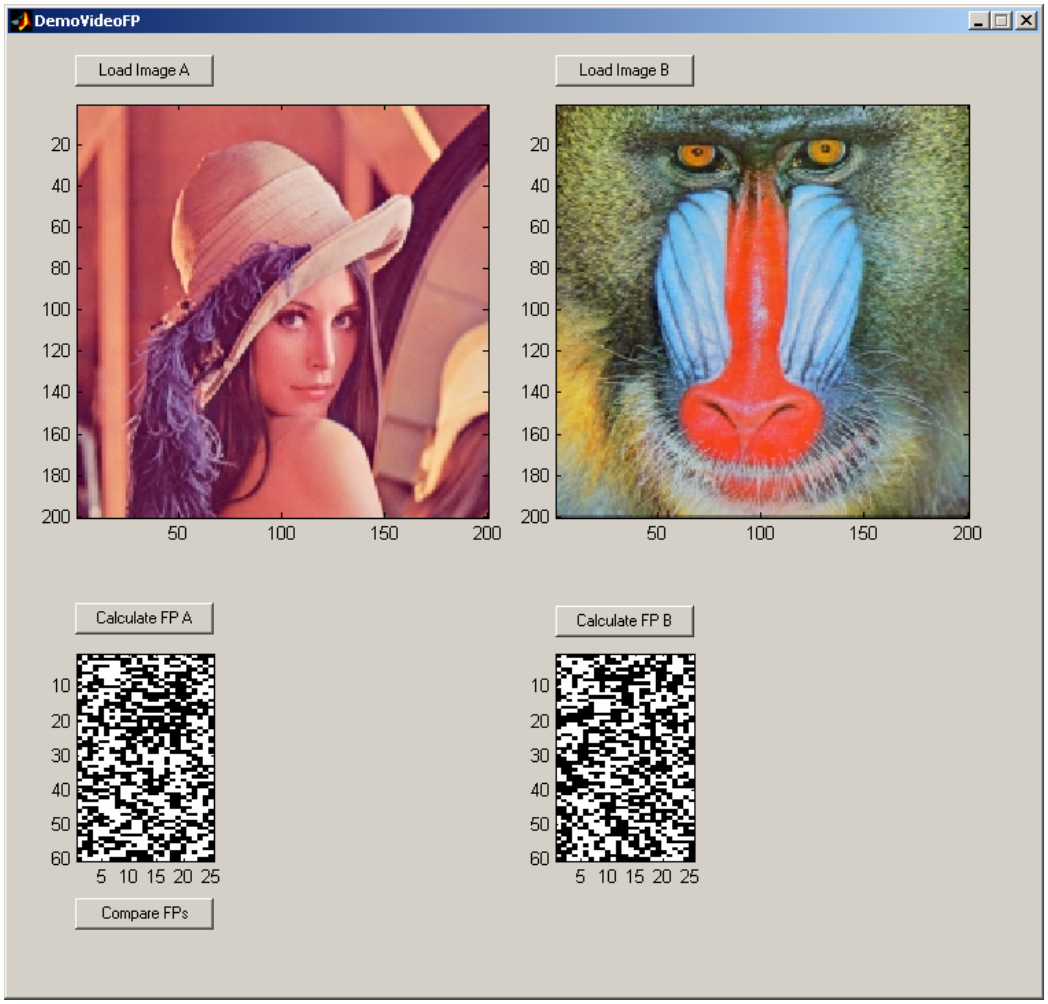
- load the reference image and the image under evaluation, to
- calculate the fingerprints for the reference image and the image under evaluation, and to
- compare the calculated fingerprints.

For the visualization of the input images and the calculated fingerprints different areas are available in the user interface.

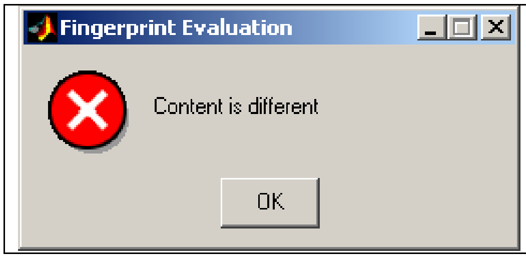
For the demonstration, the users can load selected images:



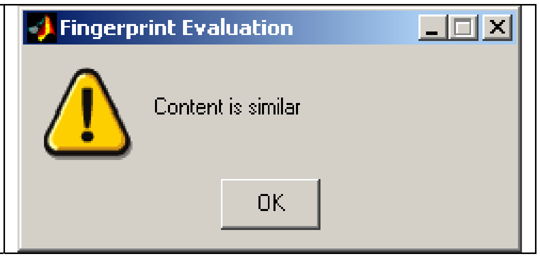
The fingerprints are calculated for the loaded images and shown visually:



After calculation of the fingerprints, the fingerprints can be compared. Dependent on the input content the fingerprints are either different (as in the above example) or similar.



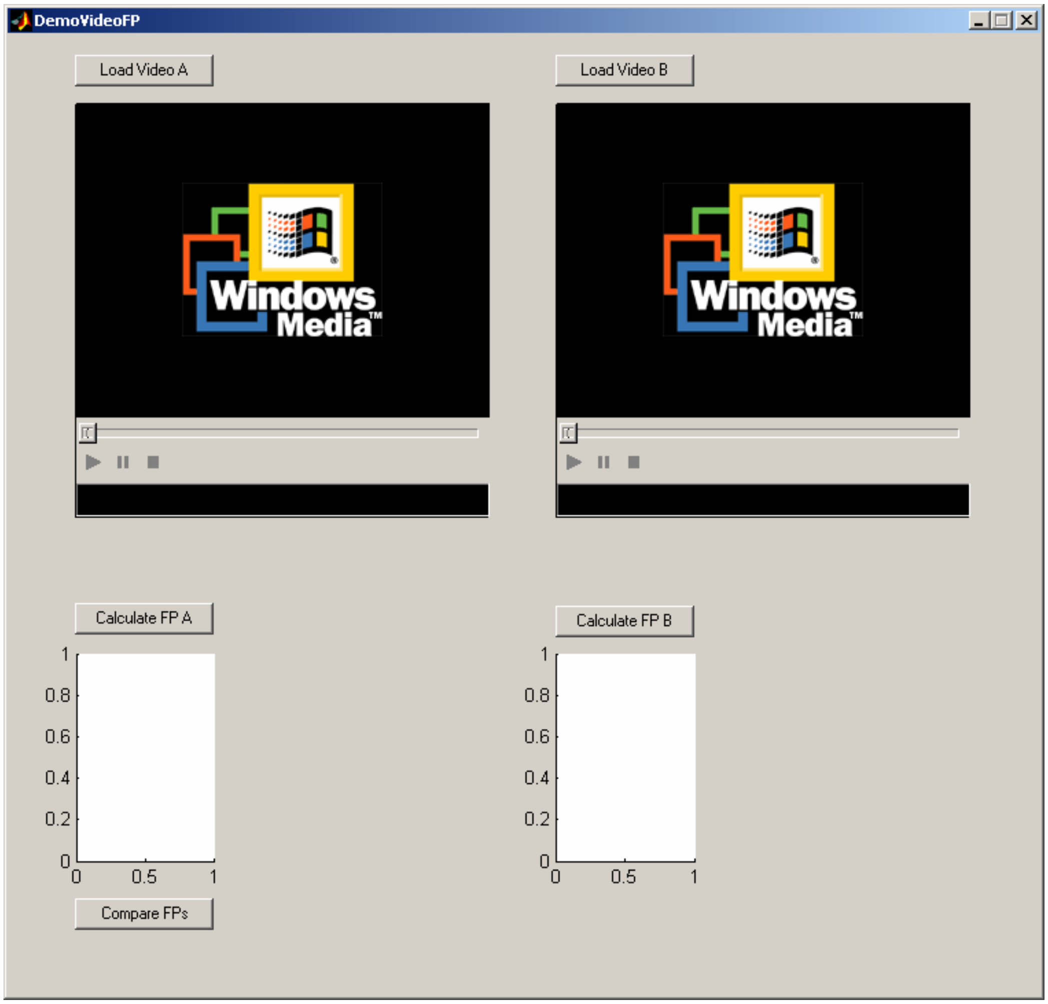
Content is different



Content is similar

C. Video Fingerprinting Demonstration

When the application is started in MatLab, a dialog is shown.

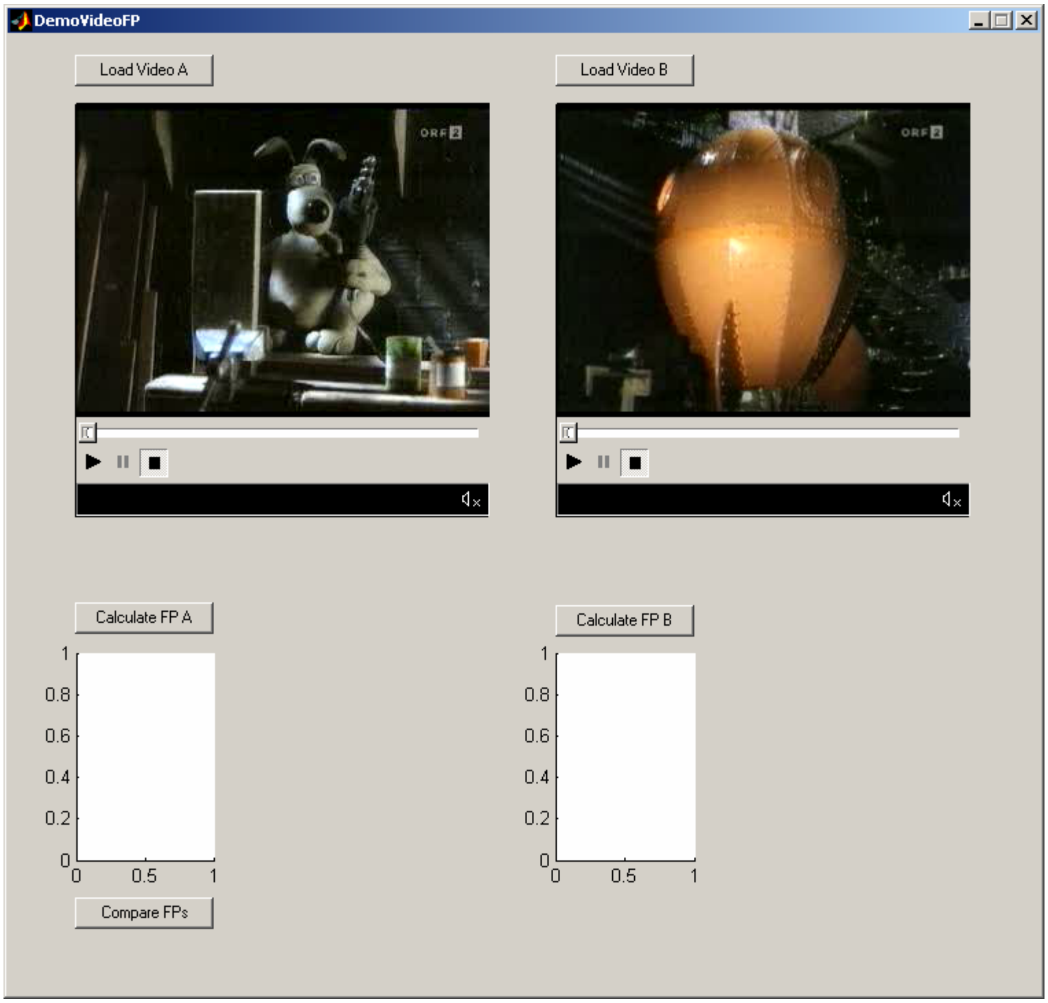


This dialog consists of four buttons, which allows to

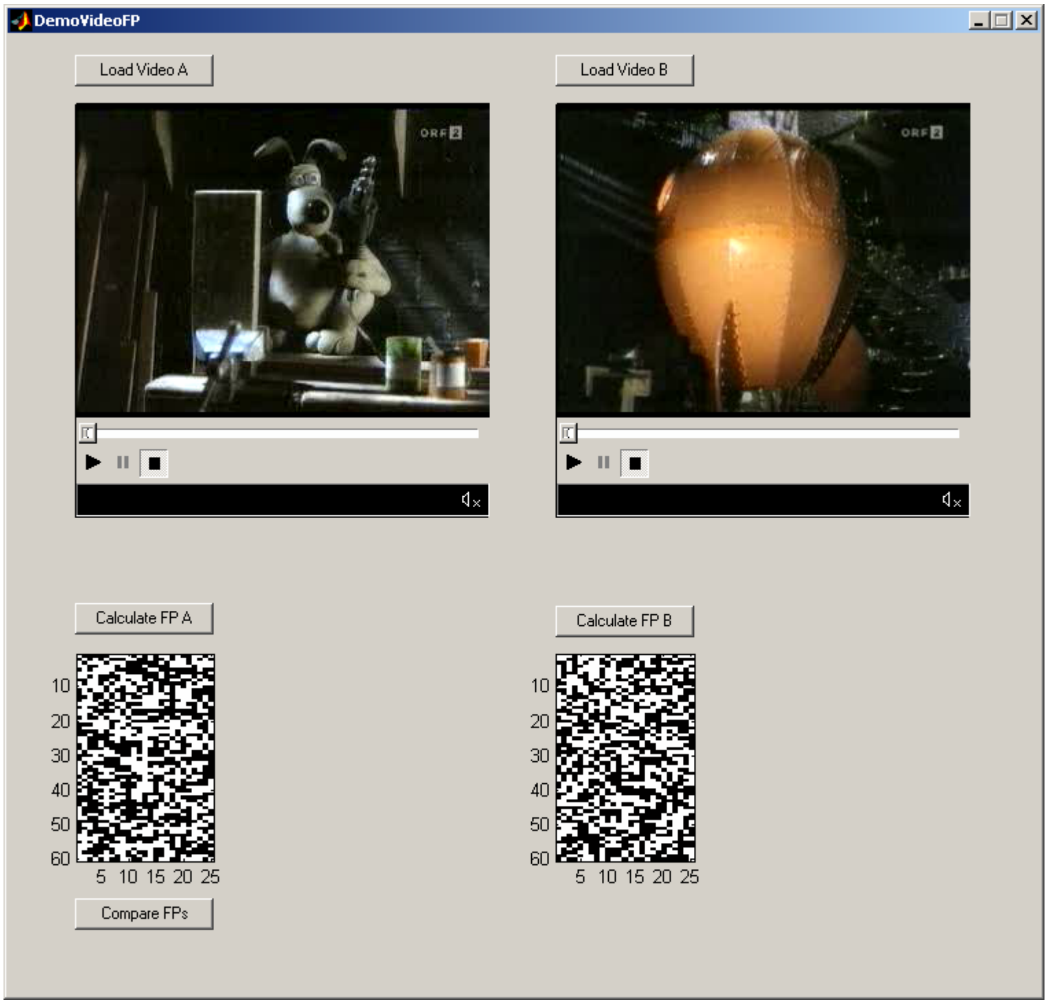
- load the reference video and the video under evaluation, to
- calculate the fingerprints for the reference video and the video under evaluation, and to
- compare the calculated fingerprints.

For the visualization of the input videos and the calculated fingerprints different areas are available in the user interface.

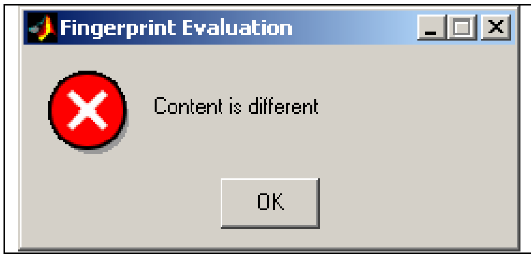
For the demonstration, the users can load selected videos:



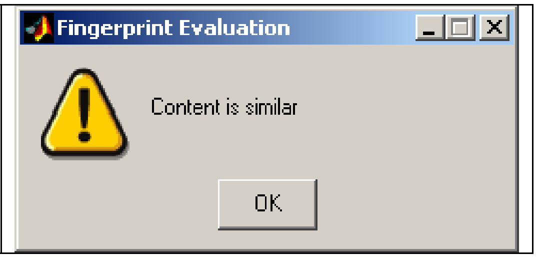
The fingerprints are calculated for the loaded images and shown visually:



After calculation of the fingerprints, the fingerprints can be compared. Dependent on the input content the fingerprints are either different (as in the above example) or similar.



Content is different



Content is similar