Using the MPEG Query Format for Cross-Modal Identification

Matthias Gruhne and Peter Dunker
Fraunhofer Institut Digitale Medientechnologie (IDMT), Ilmenau, Germany
Email: {Matthias.Gruhne,Peter.Dunker}@idmt.fraunhofer.de
Ruben Tous
Universitat Politècnica de Catalunya (UPC), Dpt. d'Arquitectura de Computadors, Barcelona, Spain
Email: rtous@ac.upc.edu

Abstract— **During the last years a vast number of multimedia databases have been established for search and retrieval of multimedia data, due to large audiovisual storage capacities and efficient compression methods. The metadata format of such databases can be based on the MPEG-7 standard. For the successful and efficient search, a query language for the interaction between client and database is crucial. One of the latest developments of the MPEG committee is the MPEG Query Format (MPQF) which is destined for the interaction between clients and MPEG-7 databases. The client may be interested to express more complex queries, which requires an interaction with several databases. Therefore an additional service provider can be used in the according environment, which accepts and understands a query formulation from a client and forwards parts of the query to one or more databases. The service provider furthermore retrieves the responses from these databases, postprocesses these results and replies a combined result list to the client.**

During the last years, the cross-modal search of visual and audio information has become more and more important. Using both domains, video and audio, turned out to be much more robust for the identification of video streams, than the visual part of the video stream alone. This paper describes a method for the audiovisual identification on remote databases using the MPQF. Additionally a service provider is deployed, which splits and aggregates the query and send them to two remote MPEG-7 databases (visual and audio) for identification. Among others a novel technique for the feature extraction on the service provider side is described, which is based on the MPQF. The interface between user and database is described in detail, examples are given and extensive results for the cross-modal search are presented.

Index Terms— **cross-modal identification, MPQF, audiovisual identification, search and retrieval**

I. Introduction

During the last years the search and retrieval of multimedia content has gained importance due to the large amount of digitally stored multimedia data. However, most audiovisual formats lack in the description of the actual content due to missing or disobeying standards. Therefore, the automatic identification of the audiovisual data based on available audiovisual databases gained importance. The interface for the communication between client and database is often based on proprietary formats established by commercial database providers. A suitable query format, which describes the search and retrieval process of multimedia data was missing. Therefore, the MPEG committee finalized a standard which defines an interface between clients (customers) and databases (service providers) for information search and retrieval. This standard is called the MPEG Query Format. During the development of this standard importance has been given to the possibility of introducing a service provider, which is placed in between client and database. This service provider is able to connect different databases, if one complex query is sent by the client. Then, the original query is analyzed and the different parts of this query are distributed to specific databases. Different multimedia search interfaces such as SQL/MM [1] have been developed during the last years. None of these interfaces supports metadata search and content based search at the same time. The query format is able to express the metadata terms, e.g., returning all movies which contain 'Bruce Willis' in the metadata and at the same time, it can be used to return all songs similar to the given one. One possible scenario for the query format is the cross-modal search of visual and audio content simultaneously [3]. Cross-modal information retrieval is a multimedia search technique, where the query, containing a certain type of media is used for searching the associated metadata. An example scenario is searching for a movie, which contains a certain soundtrack or searching for music, which appears as soundtrack in a movie. Another application for cross-modal retrieval is compensating corrupted (or absent) media sources. The combination of cross-modal methods with pre-existing single-modal retrieval methods can provide robustness and other advantages not possessed when using either of the media types alone. The novel work in this paper is the cross-modal identification of visual and audio data based on MPEG-7 in the framework of the MPQF. One of the key features hereby is the capability of query splitting by the service provider, which takes the query request and sends sub-queries to the audio and to the visual database respectively. In case the query request contains the raw data and not the MPEG-7 descriptions, the query splitter sends these data to an extraction service provider. This provider extracts an MPEG-7 compliant feature, because most databases are not capable of extracting the media themself. The service provider also aggregates the

results from different databases, which can be referred to the term Distributed Information Retrieval. The service provider uses the information sent by, e.g., an audio and a visual identification database and merges the results in order to answer with one single result to the client. This procedure is conducted with an adapted rank aggregation algorithm described in [4]. The focus in this paper is the distributed cross-modal video identification using the MPQF. A preliminary work on progress was already published in [5]. After an extensive system setup chapter and some query request examples, the results of the cross-modal search are described in detail. These results are based on a test set, containing a large number of videos with different degrees of distortion.

II. State of the Art

A. MPEG Query Format

The MPQF is an XML-based language which defines the format of queries and results, which are interchanged between clients and servers in a multimedia information search and retrieval environment. Formally, the MPQF is one part (Part 12) of the ISO/IEC 15938-12 standard, "Information Technology - Multimedia Content Description Interface" better known as MPEG-7 [6]. However, the query format is technically decoupled from MPEG-7 and can be used for querying any XML-schema based metadata. The main benefits of standardizing a language are interoperability between parties (e.g., content providers, aggregators and user agents) and platform independence; developers can write their applications involving multimedia queries independently of the database used, which fosters software re-usability and maintainability.

1) Multimedia search and retrieval: Information Retrieval (IR) and (XML) Data Retrieval (DR): Multimedia search and retrieval systems should be able to process different media with heterogeneous characteristics such as text, still and moving images, and audio. Developing such systems poses several challenges, due to the heterogeneity of the data and the fuzziness of information. One of its key aspects is the combination of Information Retrieval (IR) techniques and techniques for querying metadata (which belong to the Data Retrieval area) in the same system. Both approaches aim to facilitate users access to information, but from different points-of-view. An Information Retrieval system aims to retrieve information that are relevant to the user even though the query is not formalized, or the criteria are fuzzy. In contrast, a Data Retrieval system (e.g., an XQuery-based database) deals with a well defined data model and aims to determine which objects of the collection satisfy clearly defined conditions (e.g., the title of a movie, the size of a video file or the fundamental frequency of an audio signal).

One of the advantages of MPQF is that it allows the expression of queries combining both the expressive style of Information Retrieval and XML Data Retrieval systems (e.g., keywords and query-by-example with, e.g., XQuery). Regarding Information Retrieval, MPQF offers multiple possibilities that include but are not limited to query-by-example-media, query-by-region-of-interest, query-by-example-description, query-by-keywords, query-by-feature-range, query-by-spatial-relationships, query-by-temporal-relationships and query-by-relevance-feedback. For Data Retrieval, MPQF offers its own XML query algebra for expressing conditions over the multimedia related XML metadata, e.g., MPEG-7 or any other XML based metadata format but also offers the possibility to embed XQuery expressions.

2) MPEG Query Format language parts: The normative parts of the MPQF (see Figure 1) contain two main components: The *Query* container provides means for the communication of the search and retrieval tasks, whereas the Input query format describes the requests from a client to a multimedia information retrieval system (MMRS), which is also referred to the term database in this paper. The Output query format specifies a message container for MMRS responses to the client. The FetchResult element allows the user to request the results of a previous query. The *Management* container comprises both Input and Output messages and organizes all organizational aspects of sending a query to a database. This includes functionalities such as service discovery, service aggregation and service capability description. Note, that the term service refers to all MMRS including single databases as well as service providers administrating a set of MMRSs.

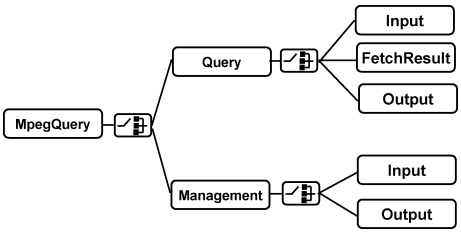

Figure 1. MPQF language parts

A query request can be composed of three different parts. A *declaration part* points to resources, (e.g., image file or its metadata description, etc.), that are reused within the query condition or the output description part. The *output description part* allows the definition of the structure as well as the content of the expectant result set by using the respective MMRS metadata description. Finally, the *query condition part* denotes the search criteria by providing a set of different query types (e.g., QueryByMedia) and expressions (e.g., GreaterThan) which can be combined by Boolean operators (e.g., AND).

In order to respond to MPQF query requests, the *Output Query Format* provides the *ResultItem* element and attributes signaling paging and expiration dates. A further description about the MPQF can be found in [7].

B. Distributed Information Retrieval and Data Integration

Search engines for multimedia or other information retrieval systems are, based [8] on a *single database* model. In this model the metadata are stored in a single database (which might be implemented centralized

or distributed), where they are indexed and conducted searchable. This model can be seen as the information retrieval version of data warehousing for data retrieval. Some information is not accessible in this model. This information can be queried but cannot be copied to the centralized database for different reasons (size, volatility, interface restrictions). The alternative is a *multi-database* model. In this model a central site (or any peer in a distributed peer-to-peer context), translates the query of the user into a standardized query, handled by an associated database. If the user performs a query, a central site translates this query and forwards it to one or several databases. The response from these databases is translated and replied to the user. This model is currently studied in the field of ”Distributed Information Retrieval”. To efficiently accomplish this task, a centralized server (the service provider) performs major operations, such as (see Figure 2):

- *Source selection*: Given the source descriptors, the user query, and maybe other statistical information, decide which sources must be queried. Source selection is required to forward the user query only to the repositories that are candidate to contain relevant documents.
- *Source querying (query splitting)*: Mapping the user query to a standardized query and forward it to one or more databases. This step is necessary in order to translate the query into one or more understandable formats.
- *Results merging (results fusion)*: Merge the ranked lists returned by the different sources. Result fusion is used to collect all retrieved documents and conveniently arrange them for presentation to the user.

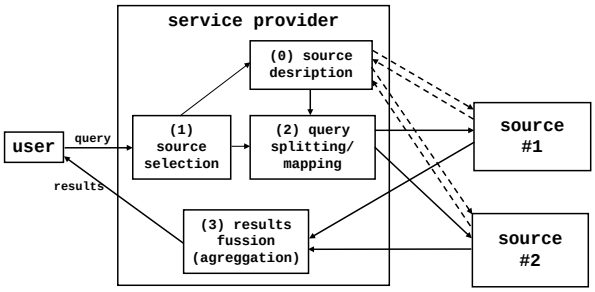

Figure 2. General Distributed Information Retrieval workflow

These operations are related to the IR aspects of multimedia search, but querying for multimedia contents combines IR and Data Retrieval techniques. In case of the data retrieval and the database research discipline, the problem of querying distributed and heterogeneous information is known as Data Integration [9] [10] [11] [12]. Traditionally, the goal of a data integration system is to provide a mediation architecture in which a user poses a query to a mediator that retrieves data from underlying sources to answer the query. The constraints of the sources access, and their potentially different data models and schemes, are the challenges of a data integration system.

In some aspects data integration and distributed information retrieval are equivalent, and in some contexts the terms are used indistinctly. However, while distributed information retrieval targets to satisfy a user query over unstructured data or semi-structured data sources, a data integration system aims to satisfy a query over also autonomous and heterogeneous, but structured data sources.

C. Audiovisual Search

Search and identification of audiovisual content has been broadly described in scientific literature in the past, e.g., [13]. Cross-modal search, however, has been considered only sparsely. Following [4], a cross-modal search strategy is defined by a combination or an interaction of different modalities. The idea in this paper is to combine audible and visual information as they are available, e.g., in movies in order to obtain suitable identification results, also if the recognition quality in one of the media decreases due to insufficient image quality or noise in the audio. However, this publication concentrates on MPEG-7 audiovisual fingerprinting technologies for media identification [6], since the MPQF is dedicated to support MPEG-7 applications. A fingerprint is a small representation of the media itself and contains features that represent a special property of a modality that is extracted by signal processing algorithms.

a) Audio Identification: A vast number of different algorithms for audio identification have been published and already evidenced as market-ready. Fundamentally different feature extraction and classification methods are described for audio identification. The extraction process can be based, e.g., on the spectral envelope, using delta features or singular value decomposition. The classification is mostly based on distances between features, e.g., [13], but also hashing methods are used [14].

Wang described an audio identification algorithm based on hash feature vectors (containing a vector with only the values 0 and 1) [15]. These vectors are calculated by extracting the spectrogram peaks (the frequency value of the most prominent part of the power spectrum of the music signal) in different frames of music. Two of these spectrogram peaks in the same frame build a pair, which is called ”combinatorial hashing”. Due to the fact that the music signal is continuous and such a pair exists per frame, a subset of these spectrogram pairs are declared as ”anchor points” and contain a time span and occurred frequency values. This information is used to create vectors of 32-bit hashes, which uniquely distinguishes millions of songs.

Haitsma et al. described another audio identification algorithm by cutting the audio signal into small pieces, each of a certain length [14]. Thereafter, a Fourier transform is applied on each of these snippets. The power spectrum is estimated and parted into 32 linearly spaced sub-bands. Within each sub-band, the values are squared and summed in order to reduce the number of dimensions. Since the fingerprint was planned to be a hash value the differences between neighbored sub-bands and between

the same band in consecutive frames is calculated. If this difference value is greater than zero, the hash value in this band is set to one and set to zero otherwise. The output of the system is a 32 bit hash fingerprint per frame. The classifier contains the hash values of all database fingerprints and identifies the query hash based on a hash table.

One kind of audio identification technology has been standardized within the MPEG-7 standard [6]. That means, the extraction algorithm is normative and has been published. Therefore, extractors can be developed independently by other parties. An MPEG-7 conform audio identification technology has been described among others in [13]. When extracting such a fingerprint, an audio signal is subdivided into several frames of a size of 30 milliseconds. After applying a hamming window function on each frame, a Fast Fourier transform is performed. Thereafter, the power spectrum is estimated and subdivided into logarithmically spaced bands. Within each band a spectral flatness measure is calculated, by dividing the geometric mean of the power spectrum values within each band by the arithmetic mean of the same values. The spectral flatness value indicates the noisiness of the tone. A value of 0 indicates noise and values if 1 specify a sinus-like tone. The computed values within each bands are referred to a feature vector and can be used for music information retrieval tasks. The raw feature vectors are further processed by combining bands and successive frames. This enables a flexible fingerprint size, depending on the requirements. If the distortion of the input signal is high, the fingerprint size is chosen larger, if the input signal quality is supposed to be good, the fingerprint size is chosen smaller, enabling a shorter classification duration. For classification, a normal nearest neighbor classifier using a Euclidean distance metric has been applied.

b) Visual Identification: There are also a number of different approaches for visual recognition described in literature. The greatest differences between these approaches are located in the feature extraction method. This method can be divided into different groups: *Frame based features* such as histograms, structure or local features; *Motion features* such as motion intensity or trajectory estimated by consecutive frames; *Features based on video shots*, e.g., shot duration or frame features of the corresponding key frames.

A collection of different types of visual features can be found in the MPEG-7 standard (visual part [6]). This standard describes also the extraction process and the format to store the features. Various publications demonstrate identification approaches based on MPEG-7 features, e.g., [16], [17]. In [18] the color layout descriptor (CLD) is used which describes the layout structure of an image. The CLD extraction is a simple and fast process. The whole image is divided into 8 by 8 pixels image regions. Within each region a dominant color is estimated, which can be the mean value. Thereafter, a discrete cosine transformation (DCT) is calculated using the 8 by 8 mean values. To reduce the computation time of the search process the number of coefficients can be reduced by a subset of the coefficients after a zigzag scan.

A typical key frame approach is presented in [19]. This algorithm estimates first key frames by means of frame differences called "Intensity of motion". Subsequently a Harris detector is applied to find interesting points and a local feature, based on Gaussian differential decompositions for each of these points is calculated. The classification is done by a nearest neighbor search.

Yuan et al. [20] presented a combination of motion features and a modification of the video shot approach. This algorithm avoids the estimation of scene changes which can cause mistakes in an early stage of feature extraction. Here, temporal segments with a fixed length are used and spatio-temporal features are taken out of these segments. Besides the feature extraction Yuan concentrates on an indexing method for classification.

During the last years, various publications focused on more efficient classification methods. Shen et al. [21] present an algorithm that concentrates on the search of complete video sequences based on a novel video summary representation. The calculation of the representation starts with a k-means clustering of similar images and the description of each cluster with its cluster position, range and density. The main idea is a reduction of the list of clusters in each video to a single numerical value that is used as a key in a fast B^+-tree. This mapping is realized by calculating a distance in the multidimensional representation space between a reference point O and the dedicated database entries O_i. The main problem can be considered as finding an optimal O to keep most of the similarity information. The O search is performed by a PCA of O_i and moving O on the first principle components out of the center of the transformed dataset. This algorithm allows also a search of near duplicates or similar video clips.

III. Distributed Cross-modal Search using the Query Format

Using the MPQF format for the cross-modal search results in the advantage, that the audio identification service and the visual identification service do not necessarily need to be situated in the same proprietary infrastructure. They can be totally independent identification services maybe even with partially other processing tasks. These databases, however, communicate with the interface defined in the MPQF and they are registered at a service provider, which aggregates the results and ensures the other cross-modal tasks. In this scenario, the core of the cross-modal system is situated at the database side. Figure 3 depicts the scenario for the cross-modal search using the MPQF in a distributed environment.

The service provider keeps a list of potential audiovisual databases. In an initialization step the client submits user preferences to the server in order to signal his priorities for the search process, e.g., that he wants only to use royalty free databases. Hence, based on this information

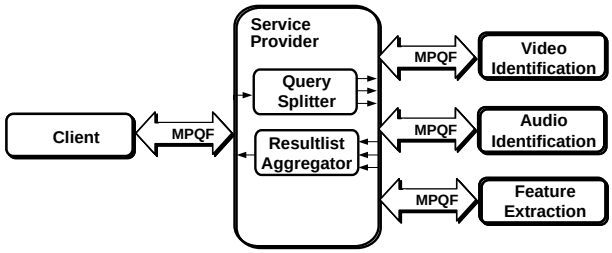

Figure 3. Cross Modal Search with the MPEG Query Format

the service provider can identify possible databases for the user's request. Some of these databases will be selected due to their capacity to process the complete query. In case of the *cross-modal search* databases can be selected due to their capability to process a partial aspect of the query, e.g., audio or visual identification. In any case, the query must be conveniently rearranged (*mapped*) in order to fit the particularities of each database. In the second case this mapping can also imply partitioning of the query functionality, which is called as *query splitting*.

During the search process, the client sends the query via the MPQF to the service provider. This query may contain an XML instance based on the MPQF and shall contain MPEG-7 descriptors for audio as well as MPEG-7 descriptors for visual information. These descriptors are extracted from, e.g., a part of a corrupted video at the client side. The service provider contains a query splitter, which separates the visual part and the audible part of the query format and sends both parts independently to the according database. The service provider contains a list of accessable databases as well as a profile of processable query elements each, since the MPQF contains a management layer, which allows the service provider to query the databases for capabilities. Separating the elements of the incoming query is easy in case of non-overlapping functionalities of the databases, meaning every database can process a different subset of tasks. In this case, the XML query expressions are separated and a number of new queries are created based on the server functionalities. In case of overlapping functionalities of the databases, the service provider needs to decide where to send the XML elements. This decision can be based on user preferences as, e.g., "Is this service for free ?" or "How reliable is this service?". The visual database returns a result list of the visual part including a confidence value for every result entry. The audio identification database returns also a result containing the audio identification results together with a confidence. The resultlist aggregator merges both result lists according to the aggregation concept described below. Furthermore it replies a merged overall result list to the client containing audio and visual references to the according database items. The focus in this paper is on the query splitter and the result aggregator.

A. Query splitting

The example in Code 1 shows an MPQF query from a user which contains two elements in the *QueryByDescription* query type, one destined to perform visual identification *mpeg7:MotionTrajectoryType* and the other destined for audio identification *mpeg7:AudioSignatureType*. These elements could originate from a video file containing both modalities. The user may want to perform an audiovisual identification, but the quality of the visual part might be unsatisfactory. By searching in both modalities, the chance of a positive identification increases because if the recognition of one modality fails, the identification is performed on the other modality.

Code 1 MPQF example query using a *QueryByDescription* query type

```
<MpegQuery>
 <Query>
  <Input>
   <QueryCondition>
    <Condition xsi:type="AND">
     <Condition xsi:type="QueryByDescription">
      <DescriptionResource resourceID="id1">
       <AnyDescription>
        <mpeg7:Mpeg7>
         <mpeg7:DescriptionUnit xsi:type="mpeg7:VideoType">
          <mpeg7:Video>
           <mpeg7:TemporalDecomposition>
            <mpeg7:VideoSegment>
             <mpeg7:VisualDescriptor
               xsi:type="mpeg7:MotionActivityType">
              <mpeg7:Intensity>2</mpeg7:Intensity>
             </mpeg7:VisualDescriptor>
             <mpeg7:VisualDescriptor
               xsi:type="mpeg7:MotionTrajectoryType">
               ...
             </mpeg7:VisualDescriptor>
            </mpeg7:VideoSegment>
           </mpeg7:TemporalDecomposition>
          </mpeg7:Video>
         </mpeg7:DescriptionUnit>
        </mpeg7:Mpeg7>
       </AnyDescription>
      </DescriptionResource>
     </Condition>
     <Condition xsi:type="QueryByDescription">
      <DescriptionResource resourceID="id2">
       <AnyDescription>
        <mpeg7:Mpeg7>
         <mpeg7:DescriptionUnit
          xsi:type="mpeg7:AudioSignatureType">
          ...
         </mpeg7:DescriptionUnit>
        </mpeg7:Mpeg7>
       </AnyDescription>
      </DescriptionResource>
     </Condition>
    </Condition>
   </QueryCondition>
  </Input>
 </Query>
</MpegQuery>
```

The client, which submits the query, the service provider and the according databases should communicate on the same interface. In case of the query format, they should all depend on the same XML schema in order to guarantee interoperability. Additionally, for the purpose of this paper, a compatibility of user, service provider and database to MPEG-7 is assumed to process the query adequately. For the cross-modal search, queries with two different kinds of requests are accepted.

1) Description Search: This method is suitable for requests, which contain MPEG-7 descriptions with features for the task of audio and visual identification. It tries to separate them and form two independent MPQF

queries, one for visual identification and the second for audio identification. A query request, containing audio and visual descriptions could look like one in (1). In order to separate this query, in a first step, the visual and the audio descriptions are retrieved from the query. Since the descriptions are assumed to be in MPEG-7, the query request is scanned for the occurrence of ”MPEG7:”. The result contains a list of available MPEG-7 descriptions. Further steps guarantee to select only the relevant parts for identification. Additionally to the necessary audio identification descriptions, further metadata, such as the duration could be contained in the stream. This information is necessarily needed for audio identification. Therefore the service provider keeps a list of the needed descriptions for each database. A visual identification database for example may only accept the MPEG-7 LayoutDescriptor Type. This type is then contained in the described list. Both lists (the list of the available descriptions of the query request and the list containing the descriptions, which each databases accepts) are then merged. For each item in the list one separate request is created, which contains a QueryByCondition request. Each request is submitted to the according database. Example 2 shows the automatically created query from the example code 1 using the description search technology of the service provider.

2) Content Search: Often the audiovisual database does not accept raw content, e.g., a video, as query, because the extraction process on the server side is very time consuming. However, the user may be interested in identifying the raw audiovisual content. In this case, the feature extraction should be conducted by the service provider. Since the service provider itself only resubmits and splits the incoming queries, the extraction process is swapped to another service provider, which is capable of extracting media content to MPEG-7 audiovisual features. This process can be also performed with the MPQF. Code 3 shows an XML stream of a request from the service provider to the extraction framework and the response in XML.

In order to start the extraction process, the MPQF request contains the OutputDescription elements, which indicates the MPEG-7 descriptions to be extracted. As condition, the ”QueryByMedia” element can be used to submit the actual video as URI or as Base64 stream to the extraction service. As result, the actual MPEG-7 descriptions are returned to the service provider via the MPQF output format. Code 4 shows an example of such a response.

The *ResultItem* element contains the extracted MPEG-7 descriptors for audiovisual identification. In a next step, new query requests are formulated, which are sent to the associated databases. In order to do so, similar techniques as explained in paragraph III-A.1 can be applied.

B. Cross-modal Aggregation

The proposed system uses an advanced cross-modal aggregation concept that consists of two basic steps: It

Code 2 MPQF example for the query from service provider to the databases

```
<!-- Request to the Video Database -->
<MpegQuery>
 <Query>
  <Input>
   <QueryCondition>
    <Condition xsi:type="QueryByDescription">
     <DescriptionResource resourceID="id1">
      <AnyDescription>
       <mpeg7:Mpeg7>
        <mpeg7:DescriptionUnit xsi:type="mpeg7:VideoType">
         <mpeg7:Video>
          <mpeg7:TemporalDecomposition>
           <mpeg7:VideoSegment>
            <mpeg7:VisualDescriptor
              xsi:type="mpeg7:MotionActivityType">
             <mpeg7:Intensity>2</mpeg7:Intensity>
            </mpeg7:VisualDescriptor>
            <mpeg7:VisualDescriptor
              xsi:type="mpeg7:MotionTrajectoryType">
              ...
            </mpeg7:VisualDescriptor>
           </mpeg7:VideoSegment>
          </mpeg7:TemporalDecomposition>
         </mpeg7:Video>
        </mpeg7:DescriptionUnit>
       </mpeg7:Mpeg7>
      </AnyDescription>
     </DescriptionResource>
    </Condition>
   </QueryCondition>
  </Input>
 </Query>
</MpegQuery>

<!-- Request to the Audio Database -->
<MpegQuery>
 <Query>
  <Input>
   <QueryCondition>
    <Condition xsi:type="QueryByDescription">
     <DescriptionResource resourceID="id2">
      <AnyDescription>
       <mpeg7:Mpeg7>
        <mpeg7:DescriptionUnit
         xsi:type="mpeg7:AudioSignatureType">
         ...
        </mpeg7:DescriptionUnit>
       </mpeg7:Mpeg7>
      </AnyDescription>
     </DescriptionResource>
    </Condition>
   </QueryCondition>
  </Input>
 </Query>
</MpegQuery>
```

Code 3 MPQF example for the extraction request.

```
<MpegQuery mpqfID="ID1">
  <Query>
    <Input>
      <OutputDescription>
        <ReqField typeName="MPEG7:AudioSignatureType"/>
        <ReqField typeName="MPEG7:ColorLayoutType"/>
      </OutputDescription>
      <QueryCondition>
        <Condition xsi:type="QueryByMedia" matchType="exact"
        preferenceValue="1">
          <MediaResource resourceID="Image001">
            <MediaResource>
              <MediaUri>http://db.mpqf.mpeg/testdata001.avi
              </MediaUri>
            </MediaResource>
          </MediaResource>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

Code 4 MPQF example for the extraction response.

```
<MpegQuery mpqfID="ID1">
  <Query>
    <Output>
      <ResultItem recordNumber="1">
        <Description
        xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
          <mpeg7:Mpeg7>
            <mpeg7:DescriptionUnit
             xsi:type="mpeg7:AudioSignatureType">
              <mpeg7:Flatness>
                <mpeg7:Vector>34 4 2 2
                </mpeg7:Vector>
              </mpeg7:Flatness>
            </mpeg7:DescriptionUnit>
          </mpeg7:Mpeg7>
        </Description>
        <Description
        xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
          <mpeg7:Mpeg7>
            <mpeg7:DescriptionUnit
            xsi:type="mpeg7:ColorLayoutType">
          </mpeg7:Mpeg7>
        </Description>
      </ResultItem>
    </Output>
  </Query>
</MpegQuery>
```

applies a small temporal resolution for the extraction of the query files in order to keep the accuracy for the comparison of the whole media files. The search results have the same small time resolution, which can be used to identify audiovisual streams. An example for that technique is the monitoring of broadcast streams of several audio channels. In this case, one feature vector is extracted every 30 ms. Another advantage of this procedure is, that the extraction and classification process can be already stopped after a small part of the whole audio song or video stream is reliably identified. The second step contains the cross-modal integration and the aggregation of the audio and visual search results to compensate missing matches for temporal excerpts of separate media streams (e.g., if the visual stream could not be reliably identified, but the audio stream or vice versa). An important prerequisite for this cross-modal processing step is the individual aggregation of the audio only and the visual only identification results. Figure 4 depicts the aggregation chain of a visual only or an audio only signal. This aggregation (done at the service-provider side) is performed as follows: At first the whole query video is divided into a number of sub-frames with a shorter duration. Then, the query splitter transmits several queries to the database, each with a fixed length and a short duration, e.g., 5, 10, 20 or 30 seconds. The identified results are collected and further post-processed. One could argue, that it would be computationally less expensive, if only a single video is send to the database and one reliable result is returned, independently from the information, if this video is identified or not. The advantage of the described method by splitting the video into subparts is, that some subparts might be identified and other subparts not, because the quality of the visual information in the video is too bad. In that way, the precise change of the reference in a continuous stream can be recognized. Furthermore, some videos may contain mash-ups of various reference streams, which can be also identified by that method.

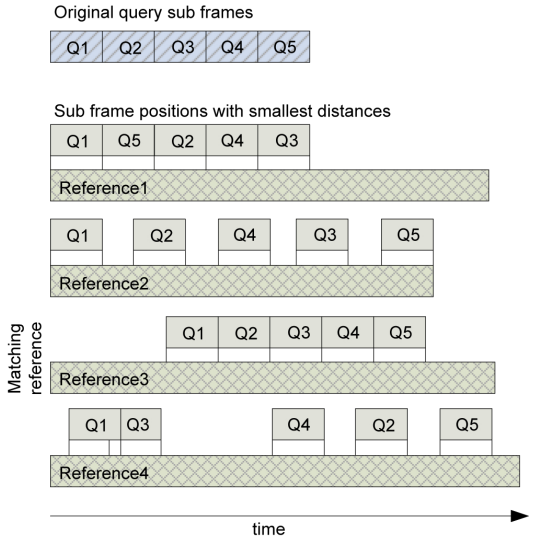

Figure 4. Search process: Q1..Q5 show the temporally consecutive query sub-frames, each with a length of, e.g., 10 seconds. Each of these Qx sub-frames is compared with every database reference (Reference 1..4) in a sliding window process. This operation returns the temporal position of the reference with the minimum distance (cityblock distance). For a matching reference candidate, all resulting query sub-frame positions shall be placed in a consecutive sequence, as depicted in Reference 3.

Each single sub-frame is individually send to the database. The database returns a result list for every sub-frame containing the cityblock distance D and a position in the reference video of the M nearest result items in the database. The results are ranked by distance D. Additionally, a confidence measure C is estimated based on the difference between the distances of the M nearest result items. This confidence ranges from 0% to 100% and indicates the probability of a match in every modality. The local position of every sub-frame is known, since the duration of each sub-frame is known (e.g., sub-frame 1 starts at second 0; sub-frame 2 starts at second 30; sub-frame 3 starts at second 60 etc.). Therefore this information can be used in order to find a final result in the database. As depicted in figure 4, a match can only be achieved, if the results of the consecutive query sub-frames match with the correct subsequent order of the reference. This behavior is the base of the aggregation concept. Within the presented work, the aggregation algorithm was extended for using multiple databases of audio and visual modalities which was realized in a cross-modal approach.

The cross-modal result list aggregator compares the identified reference time stamps of the successive sub-results for each database in order to find continuous matching paths for individual media identifiers. Figure 5 shows the result lists and the timestamps of two databases and depicts the merging process with a found final matching path. The result lists from the database contains among others the confidence c, a unique identifier id and the identified matching reference timestamps begin and

end.

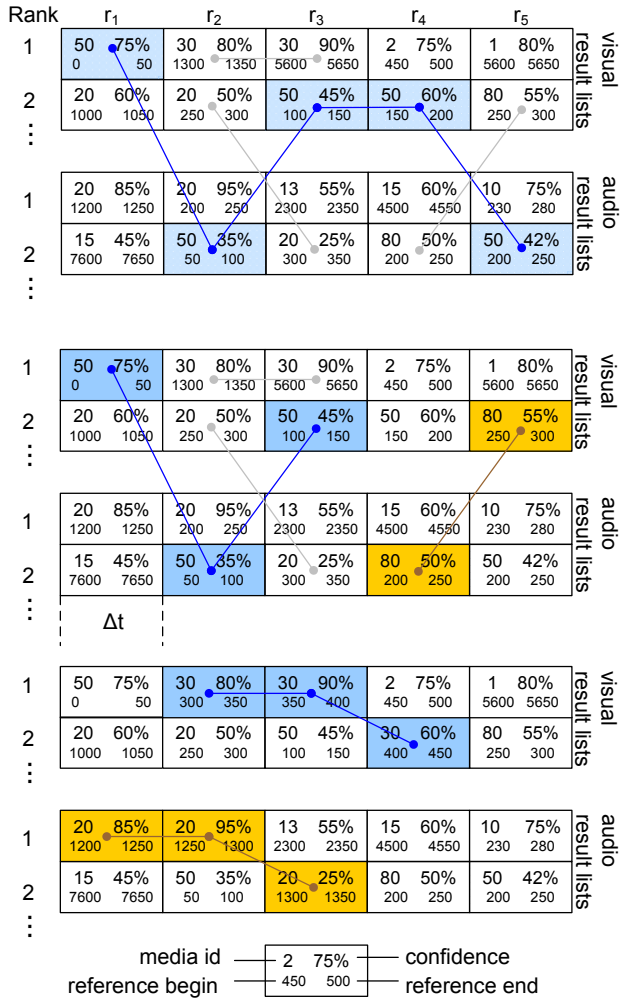

Figure 5. Search paths through audio and visual result lists. The matching references are labeled with r_x. The increasing number of x refer to the subsequent responses from the audio/visual identification.

Based on the sequence of sub-results, temporal matching paths of individual media items (ids) can now be created. Therefore the sub-result lists are stored on the service provider side. In order to generate the matching paths, the result-lists of the top N identified items are needed, since the correct item has not always the smallest distance from the reference item in the database. This can be, e.g., due to distortions of the query or because the reference item contains similar video sequences or identical audio parts. The generation of such matching paths using the time information (begin and end of the video) requires a temporal tolerance ϵ which increases the robustness of the algorithm for alignment errors evoked by re-sampling, e.g., video frame rate changes. Furthermore, small temporal scaling effects of the distorted query vs. the original can be managed.

While working with multiple databases and multiple sub-result sequences, the novelty of the described approach is the extension of already described methods to postprocess the sub-results of both modalities. The algorithm integrates the top N results of multiple databases within the same temporal slot. In order to apply this approach satisfactory, the temporal resolution of each sub-result sequence needs to be equal and has been chosen in the implementation to be five seconds per sub-frame. The first part of figure 5 depicts an application scenario for the aggregation method in the cross-modal environment by creating a matching path through audio and visual sub-result sequences.

$$P_{id} = \frac{\sum_{i=1}^{N} c_{id} \cdot \begin{cases} \alpha, & |t_{b_{id}}(i) - t_{e_{id}}(i-1)| < \epsilon \\ 1-\alpha, & \text{otherwise} \end{cases}}{N} \quad (1)$$

An additional advantage of the described method is the straightening of singular missing matches within one path through the sub-results sequences, which improves the stability of the system significantly. For applications that need to report an overall result, the reference with the longest cumulative sequence is used. But it is recommendable to assess a minimum length of such a sequence.

Previous approaches for the cross-modal aggregation used, e.g., score combination methods [22] to estimate an overall result and confidence. The aggregation algorithm of the proposed system extends these methods to the time domain and considers all scores of all sub-results. The overall confidence is calculated by a mean of all sub-result confidences with an conditioned weighting for temporal matching sub-results as depicted in equation 1.

Based on the cross-modal aggregation and the different possibilities to create the paths, a profile based selection for the user can be established as depicted in figure 5. It is possible to set up one single overall result as result as discussed above. Furthermore, a mash-up profile is possible to allow a break of the best matching path which means multiple references can be found in a single query. Finally, the simple and individual aggregation of each modality by its own can be used for an explicit search for identical audio in an audio database, e.g., searching for soundtracks and identical visual scenes for duplicate detection.

The cross-modal aggregation leads to one single recognition result for the audio and the visual fingerprint input. Code 5 shows an example of the response as it could be returned to the client. The ResultItemType contains the rank number as well as the estimated confidence. The according metadata is communicated through an MPEG-7 stream containing the name of the movie as well as the performing artist. Furthermore the exact timepoint, where the movie was found is returned to the client.

IV. Evaluation and Results

The evaluation of the different aggregation schemes: audio only, visual only or cross-modal within an information retrieval environment depends on the desired application and on its media type. The proposed distributed system focuses on information retrieval for typical TV broadcast content, such as movies or daily soaps and their potential modifications, when creating personal copies.

Code 5 MPQF example for the response from the Service Provider to the Client.

```
<MpegQuery mpqfID="ID1">
 <Query>
  <Output>
   <ResultItem xsi:type="ResultItemType"
               recordNumber="1"
               confidence="1.0">
   <Description>
    <mpeg7:Mpeg7>
     <mpeg7:DescriptionUnit
      xsi:type="mpeg7:AudioVisualSegmentType">
      <mpeg7:StructuralUnit href="">
       <CreationInformation>
        <Creation>
         <Title type="movieTitle">Cliffhanger</Title>
         <Creator xsi:type="CreatorType">
          <Role href="urn:mpeg:mpeg7:2001:PERFORMER"/>
           <Agent xsi:type="PersonGroupType">
            <Name>Sivester Stallone</Name>
           </Agent>
          </Role>
         </Creator>
        </Creation>
       </CreationInformation>
      </StructuralUnit>
      <mpeg7:MediaTime>
       <mpeg7:MediaTimePoint>T00:00:00:00F00
       </mpeg7:MediaTimePoint>
      </mpeg7:MediaTime>
     </mpeg7:DescriptionUnit>
    </mpeg7:Mpeg7>
   </Description>
  </ResultItem>
  <SystemMessage>
   <Status>
    <Code>001</Code>
    <Description>Query was successful</Description>
   </Status>
  </SystemMessage>
 </Output>
</Query>
</MpegQuery>
```

In the conducted experiments a set of 96 DVD movies has been utilized as reference database with an overall duration of more than 100 hours of video. The reference database contains also the very similar front credits and end credits of the movies as well as movies in the similar genre. Examples are ”XMen 3” and ”Superman returns” or ”The office” and ”Law and Order”.

The purpose of the evaluation consists of conducting tests with various kinds of distortions, such as transcoding, capturing content from TV screen with a camera, size downsampling and framerate adaption in the visual domain as well as different spoken languages in the audio domain. The test guarantees a mixture of approximately all potential changes that can appear, e.g., within peer2peer environments by exchanging personal copies. Therefore a set of 848 video files with a mixture of different distortions and files with significantly lower quality were used. During the evaluation phase, all 848 video files were classified in three different modes, audio-only, visual-only and cross-modal audiovisual. Furthermore, the test set contained videos without a matching reference in order to evaluate the true negatives or the rejection capabilities of the system.

The test set consists of the following manipulations:

- Visual: scale, brightness, contrast, rotate, flip, overlay by subtitles, noise, Gaussian blur, skewing, cropping, e.g., 4:3 to 16:9, framerate changes, camera capture of screen and letterbox effects
- Audible: echo, reverb, noise, highpass, lowpass, change of stereo channels
- Codec: transcoding to various compression formats with different bitrates, MPEG-1, MPEG-2, AVC, Real Media, Flash Video, Windows Media
- Further: excerpts of 5 seconds minimum, temporal combination of multiple references, trailer, items with different spoken languages

All test items were processed by each aggregation scheme: audio only, visual only and identification via the cross-modal approach. For setting up the correctly detected files TP, the correctly rejected files TN, the falsely detected files FP and the falsely rejected files FN were estimated.

In the tests, all manipulated videos were cut into several pieces and consecutively embedded into an MPQF query and sent to the service provider via the MPQF. The service provider analyzed the query as described in the last chapter and forwarded the message to the feature extraction service. After creating the audiovisual features the extraction service transmitted the query response with the fingerprints to the service provider, which processed this message and sent the audio part (the audio fingerprint) to the audio identification database and the visual part of the message (the visual fingerprint) to the visual identification database. The audio- and visual databases compared the incoming fingerprints with the internal database of the original video stream and computed identification result lists, which were replied to the service provider and cross-modal analyzed. Thereafter, one combined result was returned to the client, where the results were evaluated.

To evaluate the recognition performance the common measures *precision*, *recall* and the combined *F-measure* have been estimated:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{F-Measure} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

The results for each of the measures are depicted in table I. Interpreting the results depends on the requirement of the user. Typically, false positives are considered worse then false negatives. That could lead to the assumption, the precision is more relevant. On the other hand there could be also applications that require a higher recall and could tolerate a few false positives, e.g., semi automatic applications that utilize the results for further manual user selections. Therefore the F-Measure could be interpreted as a golden mean for various applications.

By evaluating the overall results in the first place, the most obvious discovery is the improvement of the cross-modal identification. The recall increased about 5 % while only slightly reducing the precision. That means in absolute numbers: A cross-modal comparison compared to

Approach	Precision	Recall	F-Measure
Audio	96.38 %	89.55 %	92,84 %
Visual	97.96 %	89.57 %	93,58 %
Cross-modal	96.06 %	94.50 %	95,27 %

TABLE I.
EVALUATION RESULTS OF AUDIO-ONLY AND VISUAL-ONLY ALGORITHMS AS WELL AS THE CROSS-MODAL APPROACH.

the audio-only approach results in 40 more true positives while obtaining only 4 additional false positives, which obivously increased the recognition rate significantly.

Looking at the audio only results of the proposed approach leads to the conclusion, that they are less reliable compared to other publications, e.g., [13] which describe an evaluation of only musical content. The unreliable results in the audio only tests in this publication can be explained by the fact, that the audio content contains a lot of speech and other sounds typical in movies. Additional challenges are the parts in the movies containing silence or the same video with a different spoken language, which have not been considered in this publication. The slightly better precision of the visual approach depends on the system setup and on the confidence computation, which rejects items with a small identification probability rather than having false-positives. Finally, the proposed cross-modal approach improved the recall while not significantly decreasing the precision which results in an overall improvement of the cross-modal approach of about 2 % (F-measure).

V. CONCLUSIONS

This paper described the cross-modal search on distributed systems using the MPQF. The usual cross-modal search described in literature operates on proprietary systems and mostly on the same database, whereas the described system results in the advantage, that two independent databases are also usable for other tasks and due to standardized formats, another intelligent instance like the service provider is able to distribute the queries and aggregate the results. The MPQF has been designed especially for distributed multimedia search and retrieval tasks. Its functionalities regarding source description and selection, along with its capabilities to combine Information Retrieval criteria with conditions over the XML metadata enable a cross-modal identification using MPEG-7 compliant databases. The proposed results show a significant improvement of the classification performance when using cross modal systems instead of using the individual results of audio- or visual search.

ACKNOWLEDGMENT

This work has been partly supported by the European Network of Excellence VISNET-II, PHAROS, and DIVAS, funded under the EC IST 6th Framework Program. Furthermore the publication was supported by grant No. 01MQ07017 of the German THESEUS program.

REFERENCES

[1] K. Stolze, "Sql/mm spatial: The standard to manage spatial data in relational database systems," in *Proceedings of the BTW*, 2003.

[2] D. Li, N. Dimitrova, M. Li, and I. Sethi, "Multimedia content processing through cross-modal association," *Proceedings of the eleventh ACM international conference on Multimedia*, pp. 604–611, 2003.

[3] M. Prangl, H. Hellwagner, and T. Szkaliczki, "Fast adaptation decision taking for cross-modal multimedia content adaptation," *Multimedia and Expo, 2006 IEEE International Conference on*, pp. 137–140, 9-12 July 2006.

[4] M. Gruhne, P. Dunker, M. Döller, and R. Tous, "Distributed cross-modal search with the mpeg query format," *9th International Workshop on Image Analysis for Multimedia Interactive Services*, 2008.

[5] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*. Addison-Wesley, 2002.

[6] K. Adistambha, M. Doeller, R. Tous, M. Gruhne, M. Sano, C. Tsinaraki, K. Yoon, C. H. Ritz, and I. S. Burnett, "The mpeg-7 query format: A new standard in progress for multimedia query by content," *Proceedings of the International Symposium on Communication and Information Technologies (ISCIT)*, pp. 479–484, 2007.

[7] J. Callan, "Distributed information retrieval," *Advances in information retrieval, chapter 5, pages 127-150. Kluwer Academic Publishers*, 2000.

[8] A. Y. Halevy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava, "Answering queries using views," in *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, San Jose, Calif., 1995, pp. 95–104.

[9] A. Y. Halevy, A. Rajaraman, and J. D. Ullman, "Answering queries using limited external query processors (extended abstract)," in *Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. ACM Press, 1996, pp. 227–237.

[10] A. Y. Halevy, A. Rajaraman, and J. J. Ordille, "Querying heterogeneous information sources using source descriptions," in *Proceedings of the Twenty-second International Conference on Very Large Databases*. Bombay, India: VLDB Endowment, Saratoga, Calif., 1996, pp. 251–262, see http://citeseer.ist.psu.edu/levy96querying.html.

[11] R. Pottinger and A. Y. Halevy, "MiniCon: A scalable algorithm for answering queries using views," *VLDB Journal: Very Large Data Bases*, vol. 10, no. 2–3, pp. 182–198, 2001.

[12] E. Allamanche, J. Herre, O. Hellmuth, B. Froeba, T. Kastner, and M. Cremer, "Content-based Identification of Audio Material Using MPEG-7 Low Level Description," *Proceedings of the International Symposium of Music Information Retrieval*, 2001.

[13] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proceedings of the 3rd International Symposium of Music Information Retrieval (ISMIR)*, 2002.

[14] A. Wang, "The shazam music recognition service," in *Communications of the ACM*, vol. 49, no. 8, 2006, pp. 44–48.

[15] A. Ferman, A. Tekalp, and R. Mehrotra, "Robust color histogram descriptors for video segment retrieval and identification," *Image Processing, IEEE Transactions on*, vol. 11, no. 5, pp. 497–508, May 2002.

[16] M. Bertini, A. Del Bimbo, and W. Nunziati, "Video clip matching using mpeg-7 descriptors and edit distance," *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4071, p. 133, 2006.

[17] E. Kasutani and A. Yamada, "The mpeg-7 color layout descriptor: a compact image feature description for high-

speed image/video segment retrieval," *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 674–677 vol.1, 2001.

[18] A. Joly, C. Frelicot, and O. Buisson, "Robust Content-Based Video Copy Identification in a Large Reference Database," *Image and Video Retrieval: Second International Conference, CIVR 2003, Urbana-Champaign, IL, USA, July 24-25 2003: Proceedings*, 2003.

[19] J. Yuan, L.-Y. Duan, Q. Tian, and C. Xu, "Fast and robust short video clip search using an index structure," in *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*. New York, NY, USA: ACM, 2004, pp. 61–68.

[20] H. Shen, B. Ooi, and X. Zhou, "Towards effective indexing for very large video sequence database," *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 730–741, 2005.

[21] E. Fox and J. Shaw, "Combination of multiple searches," *NIST special publication*, pp. 243–252, 1994.

Matthias Gruhne works currently at Fraunhofer IDMT, a research institute of applied technology in Ilmenau, Germany. He received his Dipl. Ing from the University of Ilmenau, Germany in 2002.

He was involved in the development of Audio Identification techniques and since 2004 he works at the MPEG standardization committee, where he contributed to the MPEG Query Format as well as to the MPEG-7 audio standard. His research interests include music information retrieval and multimodal information retrieval.

Peter Dunker received his Dipl.-Ing. in media technology at the Technical University of Ilmenau, Germany in 2003. Currently he is working at the Fraunhofer Institute for Mediatechnology (IDMT) in Ilmenau, Germany an applied research institute.

His research interests include image retrieval/classification and face detection as well as multimodal video retrieval. He was involved in the development of Video Identification techniques and consumer oriented photo retrieval projects.

Ruben Tous Ruben Tous received his his Ph.D in Computer Science and Digital Communication from UPF (Universitat Pompeu Fabra, Barcelona, Spain) in 2006. From 2000 to 2001 he worked as a consultant at CapGemini Ernst&Young in Barcelona, From 2001 to 2005 he worked at the Department of Techology of UPF. Since 2006 he is a researcher at DMAG (Distributed Multimedia Applications Group) of the Department of Computer Architecture of UPC (Universitat Politecnica de Catalunya, Barcelona, Spain) and an Assistant Professor. He is an expert for the Asociacin Espaola de Normalizacin y Certificacin (AENOR) and has been participating as spanish delegate in ISO/MPEG. His research interests include semantic-driven multimedia indexing and retrieval, knowledge representation and reasoning for multimedia understanding (multimedia ontologies) and semantic alignment.