

A Query Language for Multimedia Content *

Jonathan Mamou, Yosi Mass, Michal Shmueli-Scheuer and Benjamin Sznajder

IBM Haifa Research Lab

Mount Carmel, Haifa 31905

Email: {mamou, yosimass, shmueli, benjams}@il.ibm.com

ABSTRACT

The growing amount of digital multimedia data available today and the de-facto MPEG-7 standard for multimedia content description has led to the requirement of a query language for multimedia content. MPEG-7 is expressed in XML and it defines descriptors of the multimedia content such as audio-visual descriptors, location and time attributes as well as other metadata such as media author, media Uri and more.

While most search solutions for multimedia today are based on text annotations, having the MPEG-7 standard opens an opportunity for real multimedia content based retrieval.

In this paper we propose an IR-style query language for such multimedia content based retrieval that exploits the XML representation of MPEG-7. The query language is an extension of the "XML Fragments" query language that was originally designed as a Query-By-Example for text-only XML collections. We mainly focus on the unique characteristics of Multimedia content which needs to support similarity search query (*range search* and *K-nearest neighbors*) and queries on spatio-temporal attributes.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]

General Terms

Standardization, Languages.

Keywords

MPEG-7, Multimedia content retrieval, XML, XML Fragments, Query language

1. INTRODUCTION

* This work was partially funded by the European Commission Sixth Framework Programme project SAPIR- Search in Audio-visual content using P2P IR

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Sigir'07, MIR Workshop Jun 27, 2007, Amsterdam, The Nederland.
Copyright 2007 ACM 1-58113-000-0/00/0004...\$5.00

The rapidly increasing amount of multimedia content available all around (WWW, mobile phones, PDAs, etc) accelerates the need of standards for multimedia content description. The first step towards standardization was by developing a standard for publishing multimedia content also known as the ISO/MPEG-7¹. MPEG-7 provides a rich content description to multimedia data such as audio, video and image. It is expressed in XML and it includes Audio and Visual descriptors as well as textual metadata descriptors. Examples of MPEG-7 defined descriptors can be color and edge histograms for images, spoken lattice for speech melody for music and spatio-temporal descriptors such as location and time describing image creation.

Searching multimedia content could be done in several ways such as free text query, query by textual descriptor and of most interest is the Query-By-Example (QBE) paradigm [19] where the user supplies an example of part of a document as a query input for searching for similar documents. For multimedia content, the user can supply image, video, speech etc, as the query input. For example, using the MPEG-7 standard to search for images, visual descriptors of the query image can be compared with visual descriptors of the collection images to find similar ones.

Searching multimedia content has a great interest in the industrial and academic communities as evident by the latest kickoff of more than ten new large research projects for search in audio-visual content funded by the European Commission 6th framework program. Moreover, recently, the International Organization for Standardization has realized the urge of developing standards for searching multimedia content, and initiated a call for proposal for MP7QF² (MPEG-7 Query Format). The call includes requirements for *Query Processing*, *Query Input Format* and *Query Output Format*, in this paper we mainly focus on the *Query Input Format*.

The representation of MPEG-7 in XML and the QBE paradigm for multimedia search motivate us to use the XML query language "XML Fragments" [2, 12] as a query language for multimedia content. The main motivation behind XML Fragments was the QBE paradigm. The assumption was that similarly to full text search where the query and the document collection are given in free text, this can hold also for querying XML collections. Thus, for querying XML documents we use pieces of XML data or

¹<http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

²http://www.chiariglione.org/mpeg/working_documents/mpeg-07/mp7qf/mp7qf-req.zip

“XML Fragments” of the same nature as the documents that are queried and search for similarity between the two structures.

Since XML Fragments queries are expressed in XML, the XML representation of MPEG-7 descriptors can be used as is to embed MPEG-7 descriptors in XML Fragments queries. Still, while XML Fragments assumes text content, the comparison between multimedia features requires specific similarity search that is done by metric distance function such as Minkowski distances [1]. Moreover several similarity queries modes [18] should be supported such as *range search* and *K-nearest neighbours* (Top-k) search. Thus, XML Fragments needs to be extended to include similarity search queries for MPEG-7 features.

The purpose of this document is to define a Query Language for multimedia content that supports QBE of audio-visual features combined with text, metadata and spatio-temporal attributes. The focus of the paper is towards an IR style Query Language where returned results should be “similar” to the structure and features of the query. With respect to “similarity” on spatio-temporal attributes we mainly aim at finding results that are close to a given location or time given in the query. A future work will support more complex spatio-temporal relations between objects.

The paper is organized as follow: in section 2 we discuss some related work, in section 3 we give a short overview of MPEG-7 and in section 4 an overview of XML Fragments. Then, in section 5 we show how to extend XML Fragments to support queries over multimedia content represented as MPEG-7, followed by discussion on possible implementation issues in section 6. We then conclude with summary and future work in section 7.

2. Related Work

There have been some research works in the literature suggesting a query language for Multimedia data, we briefly describe them below and emphasize the differences from our work.

The first attempt of query language for multimedia was in the context of multimedia databases. [8, 13] extend the traditional database languages with some additions for multimedia content. However, they do not suit for the current MPEG-7 standard

The QBIC system [6] allows queries on large video and image databases based on examples images or selected colors and texture patterns. It focuses to low-level multimedia features (color, shape etc.) and follows the QBE paradigm. Despite motivations very similar to ours, QBIC does not support queries on MPEG-7 features and it does not define a formal QL, but merely focuses on the UI parts of the system.

As MPEG-7 documents are XML-based, it seems natural that extension of an existing XML query languages can be used to retrieve MPEG-7 documents. [7, 15], for example, use XQuery³ as retrieval language for accessing audiovisual content represented by MPEG-7. However, they ignore the specific numerical data representation of MPEG-7 descriptors such as color histogram, and thus fail to deal with similarity search methods that are the foundations for searching multimedia content.

The following group of works belongs to the “new generation” of MPEG-7 based query languages. Based on XML data, they tend

to propose solutions to the specific requirements of a MPEG-7 document. Defined datatypes and relationships can be effectively expressed and implicit information is taken into account. Specifically, SVQL [4] -*Semantic Views Query Language*-proposes an XQuery adaptation for MPEG-7 documents. Defined as a high level query, users can express their requirements using five different *views*: *PhysicalView*, *ProductionView*, *ThematicView*, *VisualView* and *AudioView*. SVQL has the advantage of being specially designed for the retrieval of audiovisual data. MMDOC-QL [10] proposes an XML query language with multimedia query constructs. This work, based on a logical path predicate calculus [9] defines four main clauses: OPERATION for describing the logic conclusions, GENERATE similar to the SELECT in SQL, PATTERN for describing the domain constraints (like tag, attribute, content or datatype) and the pair FROM, CONTEXT for describing multiple sources.

These works focus mainly on queries over the structure and relations expressiveness of MPEG-7 by defining powerful operators that let the user express accurate relationships constraints. However, the Descriptor specific content in an MPEG-7 document such as numerical vectors and matrices is not covered at all. As evident by [16], much of the information encoded within MPEG-7 descriptors is of this nature and not of textual nature.

VexQuery [17] presents an interesting and innovating work that is targeted to solve the weakness of XQuery to express constraints on vector-based features described in MPEG-7. This is done by an extension of the existing XQuery with two new operators *VDistanceExpr* and *VWeightDistanceExpr*. These two operators give a set of similarity measurement expressions, and this work is, thus, very close to our own solution. Still we aim at a QBE language and VexQuery, while based on XQuery, does not follow that paradigm.

3. MPEG-7 Overview

MPEG-7 emerges as a standard for describing the content of multimedia data. It is an ISO/IEC⁴ standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the well known MPEG-1, MPEG-2 and MPEG-4 standards.

Formally named "Multimedia Content Description Interface", MPEG-7 describes the multimedia content data that supports some degree of interpretation of the information meaning, which can be passed onto, or accessed by, a device or a computer code.

MPEG-7 restricts itself to few, but to the following powerful concepts [14] - *descriptors* (Ds), *description schemes* (DSs) and a *description definition language* (DDL).

Descriptors

Descriptors (Ds) are designed for describing low-level audiovisual features such as color, texture, motion and so forth, as well as high-level features of semantic objects, events and abstract objects. It is expected that most descriptors corresponding to low-level features will be automatically extracted from the multimedia

³ <http://www.w3.org/TR/xquery/>

⁴ <http://www.iso.com>

data. In Figure 1, we give an example of two MPEG-7 descriptors –ScalableColor and EdgeHistogram-

```
<VisualDescriptor type="ScalableColorType"
  numOfBitplanesDiscarded="0" numOfCoeff="64">
  <Coeff>
    -121 8 -3 87 12 14 22 37 31 13 11 3 50 14 19 21 -3 1 0 11
    -8 5 0 17 -8 2 2 4 -15 5 1 -1 1 0 0 1 0 0 1 1 6 1 1 3 1 2 4 12
    -1 0 2 2 2 3 3 -4 15 0 0 -2 1 0 -3 6
  </Coeff>
</VisualDescriptor>
<VisualDescriptor type="EdgeHistogramType">
  <BinCounts>
    0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 2 0 0 0 0 2 0 0 0 0 3 0 1 0
    0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 7 0 0 0 0 7 0 0 0 0 7 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  </BinCounts>
</VisualDescriptor>
```

Figure 1 – example MPEG-7 Visual Descriptors

Description Schemes

Description Schemes (DSs) define the structure and the semantics of the relationships between elements, both between descriptors and descriptor schemes. For example, the MPEG-7 Description Schemes in Figure 2 below specifies that the duration of a <VideoSegment> can be specified using the <Mediatime> or the <TemporalMask> element. The instantiation of the descriptions schemes can rely on automatic tools, and can require human involvement or authoring tools.

```
<complexType name="VideoSegmentType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <choice minOccurs="0">
          <element name="Mediatime"
            type="mpeg7:MediaTimeType"/>
          <element name="TemporalMask"
            type="mpeg7:TemporalMaskType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Figure 2 - example MPEG-7 schema

Description Definition Language

The *Description Definition Language* (DDL) defines the *Descriptors* and the *Descriptions Schemes*. At the 51st MPEG meeting in March 2000, it was decided to adopt the W3C’s XML Schema Language as the MPEG-7 DDL.

The adoption of XML representation for MPEG-7 motivates us to use an XML query language for querying multimedia content. Furthermore, since the MPEG-7 descriptors are already expressed in XML, we look for a QBE query language where the descriptors can be used also in the query. We found the Query-By-Example

XML Fragments an excellent candidate to be extended for querying MPEG-7 features.

4. XML Fragments Overview

The formal syntax and semantics of XML Fragments is fully defined in [2, 12] while, here, we give only a short summary.

The main motivation in defining XML Fragments was to extend classical IR system in which the query and the document collection both consists of free text. The claim is that the same can hold for XML collections so querying XML documents can be done by pieces of XML documents or “XML Fragments” of the same nature as the documents that are queried. Returned results should be not only perfect matches but also “close enough” ones ranked according to some measure of relevance.

XML Fragments are thus portions of valid XML, possibly combined with free text. For example, the following queries in Figure 2 are valid XML Fragments:

```
1. <element1 id="123">text1</element1>
2. <element1>...</element2> ...</element2>...</element1>
```

Figure 2 – XML Fragment queries

An XML Fragments query can be interpreted as a tree where each node is either an XML tag or a word. Intuitively the semantics of XML Fragments query Q is that a document D is a valid result for Q (or that Q is satisfied by D) if there is a path from Q’s root to one of its leaves that fully appears in D.

In order to allow more user control on XML Fragments and at the same time still keep their simple intuitive syntax, the following operators were added to XML Fragments:

- “+/-” prefix that can be added to elements, attributes or content. Prefixing an element with a “+” operator means that the Query subtree below that element should be fully contained in any retrieved document. Prefixing an element with “-” means that the Query sub tree below that element, should not exist in any retrieved document.
- “...” (Phrase) to enclose any free text part of the Query between quotes (“”) so as to support phrase match.
- Relation terms – for parametric search. For example, the query:


```
<book><year><.gt>2000</gt></year></book>
```

 will return all books that were published after year 2000.
- Empty tag – serves as a kind of parenthesis <> ... </> to group some query nodes together.

In a recent work [12], the language has been augmented by Boolean operators, depth constraints and the ability to define the elements to be returned (*Target Elements*).

5. XML Fragments extended for MPEG-7

As described above we would like to extend XML Fragments to support queries over MPEG-7 features. More generally, we would like to extend XML Fragments to query any XML documents that may contain some MPEG-7 nodes. Figure 3 shows an example for such a document.

```

<?xml version="1.0" encoding="UTF-8"?>
<image>
  <owner>Jo</owner>
  <title>Twilight</title>
  <description>
    Low lying fog in Norfolk countryside, twilights.
  </description>
  <comments>nice picture</comments>
  <tags>sunset sky</tags>
  <Mpeg7>
    <Description type="ContentEntityType">
      <MultimediaContent type="ImageType">
        <Image>
          <CreationInformation>
            <Creation>
              <CreationCoordinates>
                <Location>
                  <GeographicPosition>
                    <Point longitude="-36.4" latitude="18.45"/>
                  </GeographicPosition>
                </Location>
              <Date>
                <TimePoint>2007-05-15T15:44:53</TimePoint>
              </Date>
            </CreationCoordinates>
          </Creation>
        </CreationInformation>
        <VisualDescriptor type="ScalableColorType"
          numOfBitplanesDiscarded="0"
          numOfCoeff="64">
          <Coeff>-121 8 -3 ... </Coeff>
        </VisualDescriptor>
        <VisualDescriptor type="EdgeHistogramType">
          <BinCounts>0 1 ...0 5 0 </BinCounts>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
</image>

```

Figure 3 XML document with MPEG-7 features

The document has the following types of elements:

1. Metadata such as owner, title, description, comments and tags expressing semantic concepts that could be added manually or automatically to the document.
2. MPEG-7 Descriptors for location and time attributes such as Geographic position using longitude and latitude for location and TimePoint for the date..
3. MPEG-7 Visual descriptors such as ScalableColor, and EdgeHistogram.

The Metadata is defined by regular XML tags and thus can be queried by the current XML Fragments. We want to extend XML Fragments to support similarity search over the non textual fields (bolded in Figure 3) such as the date, location and visual descriptors. For those fields the query should support *Range queries* (find all elements that are less than a given distance from the query feature) and *Nearest Neighbor queries* (return the top-K

most similar elements to the query feature). In addition, it should allow weighting several features over others to boost their importance in the final scoring of results. For example, the query should allow for giving higher weight to a color histogram over an edge histogram.

We describe below the extensions to XML Fragments to support those requirements.

5.1 Querying metadata and free text

The current XML Fragments already supports querying Metadata. For example:

The free text query ‘twilight’ will retrieve the above document, since the term ‘Twilight’ appears in it.

Similarly, the query ‘Twilight pinkish Norfolk’ will retrieve the above document, since the three terms Twilight, pinkish and Norfolk appear in it.

The query ‘<title>Twilight</title>’ will also retrieve the above document since it contains both the queried title and the queried comment.

However, the query ‘<title>Norfolk</title>’ will not retrieve the above document: Norfolk appears in the document but not under the <title> tag.

5.2 Querying MPEG-7 Descriptors

To query the MPEG-7 descriptors, we introduce a new XML Fragments query tag, <Mpeg7Query>, that can be placed anywhere in an XML Fragments query. The tag is defined by the XML Schema in Figure 4 below.

The <Mpeg7Query> tag has one child and two attributes. The child, defined by the <sequence><any.../</sequence> part can be any valid MPEG-7 node but usually it will be a node that is designated for some similarity search such as e.g. a VisualDescriptor. An implementing search engine should decide on the appropriate search code to apply to that feature.

We further assume that the Query is run by a Query Processor through some API and one of the parameters to the Query Processor is K - the number of results to retrieve. It is then up to the Query Processor to propagate this K (or some other value) to the appropriate code that handle each such feature to perform K-NN query on that child. To support ‘range’ queries we introduce the “range” attribute that contains a positive value specifying the range that is used for similarity search over the feature

It is worth noting that in most cases the described query syntax will be generated by some UI or by some automatic tools for feature extraction. For example it is possible that the query is given by the user as an image with some metadata and there is some code that translates it into the proposed syntax.

An advantage of XML fragments is that it allows combining general metadata tags with several <Mpeg7Query> fragments. For instance, query may involve color histogram, shape and location and it may be useful to define weights for the different parts of the query. For example, a user may want to give a color feature more weight than a shape feature. To support this functionality we introduce the attribute “weight” that contains a strictly positive value specifying the weight given to that feature. The weight attribute is optional and if missing is defaulted to 1 like the weight of all other query parts that are not under an Mpeg7Query tag.

Lets denote by 'w' the value of the weight attribute; $w > 1$ boosts the score of its children by 'w' while $w < 1$ decreases the score of its children by 'w'.

```
<element name="Mpeg7Query">
  <complexType>
    <sequence>
      <any minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="range" use="optional">
      <simpleType>
        <restriction base="float">
          <minInclusive value="0.0"/>
        </restriction>
      </simpleType>
    </attribute>
    <attribute name="weight" use="optional" default="1.0">
      <simpleType>
        <restriction base="float">
          <minExclusive value="0.0"/>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>
</element>
```

Figure 4 - Schema definition for Mpeg7Query

Note that the range and weight attributes are optional so the defaults for each Mpeg7Query tag are weight=1 and top-K query.

The semantic of the <Mpeg7Query> tag is to retrieve the top-K (K is a query API parameter) similar documents with respect to the embedded MPEG-7 descriptor. If a range is defined then the semantics is to retrieve the top-K similar documents with respect to the given MPEG-7 descriptor in the given range.

For example, the query in Figure 5 will retrieve the top-K documents such that their feature ScalableColor is at most 17 units (according to some distance function) from the ScalableColor feature described in the query.

```
<Mpeg7Query range="17">
  <VisualDescriptor type="ScalableColorType"
    numOfBitplanesDiscarded="0"
    numOfCoeff="64">
    <Coeff>11 92 -3 87 ... -3 -6</Coeff>
  </VisualDescriptor>
</Mpeg7Query>
```

Figure 5 - querying by VisualDescriptor

The same query, but without the range attribute field, will retrieve the top-K most similar documents in their ScalableColor feature (according to some distance function) from the queried ScalableColor.

Spatio-temporal attributes, like location and time, can be queried using the same syntax. For example, the query in Figure 6 below retrieves the top-K documents which include an object with creation date similar to the query date March 25, 1998 at 9P.M., three minutes, thirteen seconds and 10/30 of a second. Like in the

VisualDescriptor case, the similarity is defined by some distance function defined for dates.

```
<CreationCoordinates>
  <Mpeg7Query >
    <TimePoint>
      1998-03-25T21:03:13:10F30
    </TimePoint>
  </Mpeg7Query>
</CreationCoordinates>
```

Figure 6- querying by creation date

It should be noted that this last query involves both similarity search and XML structure constraint. A valid resulted document should have a "similar" <TimePoint> descriptor to the one given in the query but also it should satisfy the constraint that its "similar" <TimePoint> descriptor must be under the <CreationCoordinates> tag. Moreover, by the definition of XML Fragments, the <CreationCoordinates> can be any ancestor of the <TimePoint> and not necessarily its direct parent. This gives more flexibility since there is no need to specify all tags hierarchy in a query. We give in section 6 a possible implementation for checking combination of both structural and "similarity" based constraints.

XML Fragments can be also used to express complex queries involving several metadata and MPEG-7 descriptors. For example, the query in Figure 7 will retrieve the top-K documents that have as title the keyword Twilight and that are similar in the ScalableColor feature and that have a <Location> tag that is close (by some location distance function) to the queried GeographicPosition . Since the query in Figure 6 does not have weights then it is up to the scoring implementation of the query processor to decide how to combine the similarity results returned from the three query parts (metadata, visualdescriptor and location) into a single sorted result list.

```
<title>Twilight</Title>
<Mpeg7Query>
  <VisualDescriptor type="ScalableColorType"
    numOfBitplanesDiscarded="0"
    numOfCoeff="64">
    <Coeff>11 92 -3 87 ... -3 -6</Coeff>
  </VisualDescriptor>
</Mpeg7Query>
<Mpeg7Query>
  <GeographicPosition>
    <Point longitude="-34.7" latitude="19.75"/>
  </GeographicPosition>
</Mpeg7Query>
```

Figure 7 - Complex query

In the case of image retrieval, it is evident [3] that effectiveness of the results can be improved by combining multiple Visual descriptors in a single query. For example the query in Figure 8 below shows a combination of ScalableColor and EdgeHistogram with appropriate weights. The query looks for images similar in their SclabaleColor and EdgeHistogram to the features defined by the query. It should be noted that each such feature may be indexed using a different metric space and thus we can define different range values to the different features. In the example

below we look for images that are in range 17 according to the ScalableColor distance function and are in range 12 according to the EdgeHistogram distance function.

The example also demonstrates the use of the query “weight” attribute. The ScalableColor feature is given a weight 2.0 which is twice the default weight (1.0) given to the EdgeHistogram feature.

```
<Mpeg7Query range="17" weight="2">
  <VisualDescriptor type="ScalableColorType"
  ...
  </VisualDescriptor>
</Mpeg7Query>
<Mpeg7Query range="12">
  <VisualDescriptor type="EdgeHistogramType"/>
  ...
  </VisualDescriptor>
</Mpeg7Query>
```

Figure 8 Query with weights

5.3 Multi modal queries

The proposed query language can be used to write multi modal queries, i.e., queries combining features extracted from different kinds of multimedia, for example queries combining image and speech. Consider for example the XML file in figure 9 below.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageWithAudio>
  <image>
  ....
  </image>
  <audio>
    <Mpeg7>
      <DescriptionUnit type="SpokenContentLatticeType">
        <Block audio="speech" num="1">
          <Node num="1" timeOffset="2550">
            <WordLink word="glasses" probability="0.508"/>
            <WordLink word="graphic" probability="0.279"/>
            <WordLink word="grass" probability="0.213"/>
          </Node>
          <Node num="2" timeOffset="2760">
            <WordLink word="on" probability="1"/>
          </Node>
          <Node num="3" timeOffset="2920">
            <WordLink word="my" probability="1"/>
          </Node>
          <Node num="4" timeOffset="3050">
            <WordLink word="screen" probability="0.908"/>
            <WordLink word="spline" probability="0.092"/>
          </Node>
          <Node num="5" timeOffset="3130"/>
        </Block>
      </DescriptionUnit>
    </Mpeg7>
  </audio>
</ImageWithAudio>
```

Figure 9 - an XML file with image and speech

The file contains two parts: an <image> part containing meta- data and visual descriptors as in Figure 3 above and an <audio> part containing transcription of a spoken data that is associated to the image. Such a combination can be generated by today’s cell

phones that have the capability to take a picture and to record some speech annotations.

The <audio> part is described by the Mpeg7 SpokenContentLatticeType that contains a lattice of spoken words arranged in <nodes>. Each node can have several options for the transcribed words each with a detection probability assigned by the speech-to-text algorithm. Such lattice information can be used for information retrieval on the speech data as described e.g. by [11].

The query in Figure 10 below can be used to retrieve the above document. The query contains a VisualDescriptor with weight 2.0 and a SpokenContentLattice with weight 0.5. The query should return documents that contain both features such as e.g. the document in Figure 9 above.

```
<Mpeg7Query weight="2">
  <VisualDescriptor type="ScalableColorType"
  ...
  </VisualDescriptor>
</Mpeg7Query>
<Mpeg7Query weight="0.5">
  <DescriptionUnit
    type="SpokenContentLatticeType">
    ...
  </DescriptionUnit >
</Mpeg7Query >
```

Figure 10 Multi-modal query

It should be noted that while giving the VisualDescriptor weight that is four times the weight of the speech part, it is still possible that returned results will contain an image that is very similar to the given image but without a match in the speech part. To force both parts to exist in a document one can prefix both descriptors by a ‘+’ thus forcing all results to have a non zero similarity with both the visual and speech features.

6. A Possible Implementation

In this section we describe a possible implementation for our proposed Query Language. The proposed implementation is not meant to be efficient but just to demonstrate that the constraints defined by the query language can be easily implemented by a search engine.

To be able to query XML documents that contain MPEG-7 descriptors, an adequate indexing schema must be assumed. We assume therefore

1. A text based XML index that can answer pure XML Fragments queries.
2. MPEG-7 specific indices for descriptors that require specific similarity search methods [18]. For example, the document in Figure 3 contains 2 spatio-temporal features and 2 visual descriptor features.

Indexing an XML document is done by sending the structure and text into the text XML index and sending each special feature (that requires similarity search) into an appropriate index that support such similarity search. For simplicity let us assume that each feature in each of the indices is indexed with some payload information containing its original document id and its character

offset in the document. For example the document in Figure 3, the tag <GeographicPosition> is indexed into the text XML index with offset 220 and the spatial feature <Point> is indexed into the spatial index with offset 246, further, the visualDescriptor ScalableColorType is indexed with offset 437.

At query time, the query is parsed and decomposed to text part that includes all the text including the query structure and to sub queries that correspond to specific features under the <Mpeg7Query> tags. For example, the query in Figure 11 below is decomposed into two sub queries.

```
<Image>
  <title>Twilight</Title>
  <Mpeg7Query >
    <VisualDescriptor type="ScalableColorType"
      numOfBitplanesDiscarded="0"
      numOfCoeff="64">
      <Coeff>11 92 -3 87 ... -3 -6</Coeff>
    </VisualDescriptor>
  </Mpeg7Query>
</Image>
```

Figure 11 - Query ScalableColorType

One contains only the text and structure –

```
<Image>
  <title>Twilight</Title>
```

and the other contains the feature part of ScalableColorType that should also be under the same <Image> tag -

```
<VisualDescriptor type="ScalableColorType"
  numOfBitplanesDiscarded="0"
  numOfCoeff="64">
  <Coeff>11 92 -3 87 ... -3 -6</Coeff>
</VisualDescriptor>
```

The query processor sends the ScalableColorType part to the index that handles ScalableColor feature, and it returns a list of results sorted by their distance from the given query where each result contains document id and offset in that document. Similarly, the text part is executed on the XML text index and it also returns a list of results sorted by similarity to the queried text. The Query Processor then needs to merge the two result lists and return only documents that match both query parts. This can be done by detecting identical document ids in both result lists and for each document id check the offsets of the matches to verify that the requested query structure is satisfied.

It should be noted that the above naïve implementation does not deal with optimization issues such as how many results to retrieve for each sub query to achieve enough final valid results but this is outside the scope of this document.

7. Conclusions and Future Work

In this paper, we presented a Query-By-Example language for IR style search in Audio-Visual MPEG-7 content where both the collection and the queries are express in XML. The proposed

syntax exploits the XML representation of MPEG-7 descriptors and supports (weighted) similarity search queries, comprising both *range* and *top-K* queries for any combination of valid MPEG-7 descriptors combining both Audio-visual descriptors, text and spatio-temporal attributes. The syntax extends the XML Fragments with a new MPEG-7 defined tag <Mpeg7Query> and is extensible to support queries over any MPEG-7 descriptors.

The work presented here answers some of the important challenges pose by the ISO MP7QF⁵ call for MPEG-7 Query Input Format. It does not intend to provide a complete solution, however, out of the above call it does answer the query-by-textual description, free text query, QBE paradigm, query-by-descriptions specified by MPEG-7 standard, limiting the size of the result set, etc. Finally, it is a pure XML language.

A possible extension can be to add support for database oriented spatio-temporal operators such as spatial relationship of objects and/or temporal relationship of objects.

8. REFERENCES

- [1] P. E. Black, "L_m distance", in *Dictionary of Algorithms and Data Structures*, Paul E. Black, ed., [U.S. National Institute of Standards and Technology](http://www.nist.gov/special_services/standards/technology/).
- [2] A. Broder, Y. Maarek, Y. Mass, and M. Mandelbrod. Using XML to Query XML - From Theory to Practice. In *Proceeding of RIAO*, 2004.
- [3] H. Eidenberger, "How good are the visual MPEG-7 features?", SPIE & IEEE Visual Communications and Image Processing Conference, Lugano, Switzerland, 2003
- [4] N. Fatemi, O. Khaled, and G. Coray. An XQuery adaptation for MPEG-7 documents retrieval. In *XML conference and Exposition*, 2003.
- [5] N. Fatemi, M. Lalmas, and T. Roelleke. How to retrieve multimedia documents described by MPEG-7. In *Semantic Web and Information Retrieval*, 2004.
- [6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC System. *Computer*, 28(9):23{32, September 1995.
- [7] J.-H. Kang, C.-S. Kim, and E.-J. Ko. An XQuery engine for digital library systems. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 400-400, Washington, DC, USA, 2003. IEEE Computer Society.
- [8] J. Li, M. Ozsu, and D. Szafron. MOQL: A multimedia object query language, 1997.
- [9] P. Liu, L. H. Hsu, and A. Chakraborty. Path predicate calculus: towards a logic formalism for multimedia XML query languages. *Markup Lang.*, 3(1):93-106, 2000.
- [10] P. Liu and L. Sushu. Queries of digital descriptions in MPEG-7 and MPEG-21 XML documents. In *XML Europe 2002*, Barcelona, Spain, 2002.

⁵http://www.chiariglione.org/mpeg/working_documents/mpeg-07/mp7qf/mp7qf-req.zip

- [11] J. Mamou, B. Ramabhadran, O. Siohan. "Vocabulary independent spoken term detection". In *Proceedings of SIGIR*, 2007.
- [12] Y. Mass, D. Sheinwald, B. Sznajder, and S. Yogev. XML Fragments extended with database operators. In *Proceeding of RIAO*, 2007.
- [13] J. Melton and A. Eisenberg. SQL multimedia and application packages (SQL/MM). *SIGMOD Rec.*, 30(4):97-102, 2001.
- [14] Philippe Salembier and John Smith – Chapter 6 in "Introduction to MPEG-7: Multimedia Content Description Interface", B. S. Manjunath, Philippe Salembier and Thomas Sikora editors, Wiley pub., April 2002
- [15] D. Tjondronegoro and Y.-P. P. Chen. Content-based indexing and retrieval using MPEG-7 and XQuery in video data management systems. *World Wide Web*, 5(3):207-227, 2002.
- [16] U. Westermann and W. Klas. An analysis of XML database solutions for the management of MPEG-7 media descriptions. *ACM Comput. Surv.*,35(4):331-373, 2003.
- [17] L. Xue, C. Li, Y. Wu, and Z. Xiong. VeXQuery: An XQuery Extension for MPEG-7 Vector-based Feature Query. In *Proceedings of SITIS*, 2006.
- [18] P. Zezula, G. Amato, V. Dohnal, M. Batko. "Similarity search: The Metric Space Approach". 2006, XVII, 220 p., Hardcover ISBN: 978-0-387-29146-8
- [19] M. M. Zloof. Query by example. In *AFIPS NCC*, pages 431-438, 1975.