

# Standards for Query Formalization in Mobile Visual Search

Ruben Tous and Jaime Delgado

Universitat Politècnica de Catalunya (UPC BarcelonaTech), Barcelona, Spain  
rtous@ac.upc.edu, jaime.delgado@ac.upc.edu

**Abstract.** The research around visual search is gaining relevance due to the evolution in the generation and usage of digital images. A significant push comes from the mobile visual search topic, due to the widespread proliferation of camera enabled mobile devices. The new scenarios are increasing the urgency of novel solutions for challenging problems such as the efficient coding of compact visual descriptors and the interoperability of distributed visual search query interfaces. Currently, almost every visual search service offers a different retrieval interface and image metadata description format, preventing unified and efficient access. In this context, standardization groups such as ISO/IEC SC29/WG11 (MPEG) and ISO/IEC SC29/WG1 (JPEG) have been working to create unified interfaces for image repositories. In one hand, MPEG provides the ISO/IEC 15938-12 (MPEG Query Format, MPQF), which standardizes a query language for multimedia repositories and has also started an activity for standardizing compact descriptors for visual search (CDVS). On the other hand, JPEG is now finishing the ISO/IEC 24800 (JPSearch), which provides solutions to the image metadata interoperability problem. This paper analyzes how these standardization activities can be combined to satisfy the requirements posed by the mobile visual search scenario, which are their limitations and which are the necessary actions to be taken by the standardization committees in order to overcome them.

**Key words:** visual search, cbir, metadata, interoperability, image, information retrieval, standards, jpsearch, mpqf

## 1 Introduction

Visual search allows overcoming the inherent limitations of text-based information retrieval systems. Relying on structured or unstructured text-based annotations is impractical for very large image databases, or for images that are generated continuously or ephemerally (e.g. from surveillance cameras, medical devices or smartphones). The increasing amount of digitally stored images and the widespread proliferation of camera enabled mobile devices, such as mobile phones, PDAs or tablets, poses new challenges to researchers and practitioners in this area. One of the problems to be addressed is the one related to query interface interoperability. There are already multiple existing mobile visual search

systems but every one provides a different search interface and multimedia metadata description format. This fact prevents users from experiencing a unified access to the repositories. Systems aiming to provide a unified query interface to search images hosted in different systems without degrading query expressiveness need to address several questions which include but are not limited to the following:

- Which mechanism is going to be used for query formalization? How is this mechanism going to cope with visual search queries?
- Which metadata schema or schemas are going to be exposed to user queries? Is the system going to expose a mediated/pivot schema?
- How the mappings among the underlying target metadata schemas are going to be generated?
- Which formalism is going to be used to describe the mappings?
- How is the system going to use the mappings during querying?

Currently many standardization efforts are trying to provide answers to some of these questions. Two of the most relevant initiatives are the ISO/IEC 15938-12:2008 (MPEG Query Format or simply MPQF) [1, 2, 3] and ISO/IEC 24800 (JPEG’s JPSearch framework) [4, 5]. While MPQF offers a solution for the query interface interoperability, JPSearch (whose Part 3 makes use of MPQF) faces the difficult challenge to provide an interoperable architecture for images’ metadata management. On the other hand, MPEG has also started an activity for standardizing compact descriptors for visual search (CDVS) [6]. This standard will unify the way to exchange image summaries in visual search queries. Developers of mobile visual search applications willing to provide a standard interface will face the problem of selecting the proper components of each one of these standards and try to figure out how to combine them. This paper analyzes how these three standardization activities can be combined to satisfy the requirements posed by the mobile visual search scenario, which are their limitations and which are the necessary actions to be taken by the standardization committees in order to overcome them.

## **2 ISO/IEC 15938-12:2008 standard (MPEG Query Format or MPQF)**

A key element in all the different approaches to distributed image search&retrieval is the interchange of queries and API calls among all the involved parties. The usage of different proprietary interfaces for this task makes extremely difficult the deployment of distributed image search services without degrading the query expressiveness. The progressive adoption of a unified query interface would greatly alleviate this problem. In december 2008 the MPEG standardization committee (ISO/IEC JTC1/SC29/WG11) released a new standard which provides a standardized interface to search functionalities in distributed multimedia databases (including not only still images but also video and audio), the MPEG Query Format (MPQF). To achieve this goal, the MPQF standard specifies precise input

and output parameters to express multimedia requests and uniform client side processing of result sets, respectively. Essentially, MPQF is an XML-based query language that defines the format of the queries and replies exchanged between clients and servers in a distributed multimedia search-and-retrieval system. In one hand, standardizing such kind of language fosters *interoperability* between parties in the multimedia value chain (e.g. content providers, aggregators and user agents). On the other hand, MPQF favors also *platform independence*; developers can write their applications involving multimedia queries independently of the system used, which fosters software reusability and maintainability. Figure 1 shows a diagram outlining the basic MPQF scenario. In the simplest case, the requester may be a users agent and the responder might be a retrieval system. However, MPQF has been specially designed for more complex scenarios, in which users interact, for instance, with a content aggregator. The content aggregator acts at the same time as responder (from the point-of-view of the user) and as a requester to a number of underlying content providers to which the user query is forwarded.

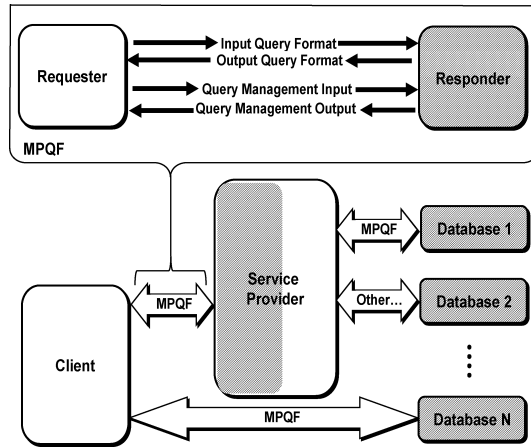


Fig. 1: MPEG Query Format diagram

## 2.1 MPEG Query Format syntax and terminology

MPQF queries (requests and responses) are XML documents that can be validated against the MPQF XML schema (see Figure 2). MPQF instances include always the *MpegQuery* element as the root element. Below the root element, an MPQF instance includes the *Query* element or the *Management* element. MPQF instances with the *Query* element are the usual requests and responses of a digital content search process. The *Query* element can include the *Input* element or the *Output* element, depending if the document is a request or a response. The part of the language describing the contents of the *Input* element (requests) is

named the Input Query Format (IQF) in the standard. The part of the language describing the *Output* element (responses) is named the Output Query Format (OQF) in the standard. IQF and OQF are just used to facilitate understanding, but do not have representation in the schema. Alternatively, below the root element, an MPQF document can include the *Management* element. Management messages (which in turn can be requests and responses) provide means for requesting service-level functionalities like interrogating the capabilities of a MPQF processor.

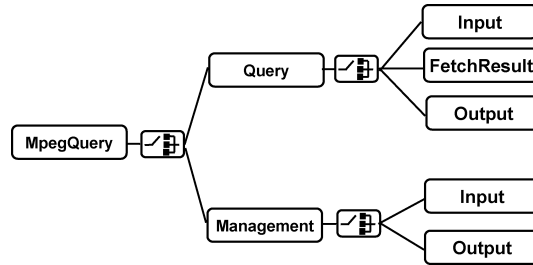


Fig. 2: MPQF language parts

Example in Code 1 shows an input MPQF query asking for JPEG images related to the keyword "Barcelona".

## 2.2 MPEG Query Format database model

As happens with any other query language, an MPQF query is expressed in terms of a specific database model. The MPQF database model formally defines the information representation space which constitutes the evaluation basis of an MPQF query processor. MPQF queries are evaluated against one or more multimedia databases which, from the point-of-view of MPQF, are unordered sets of *Multimedia Contents*. The concept of *Multimedia Content* (MC) refers to the combination of multimedia data (resource) and its associated metadata. MPQF allows retrieving complete or partial MC's data and metadata by specification of a condition tree. So, MPQF deals with a dual database model (see Figure 3) constituted by content and metadata.

Example in Figure 4 shows a graphical representation of an MPQF condition tree carrying two different conditions which reflect the duality of the database model used by the language. In one hand, there's an CBIR-like condition using the *QueryByExample* query type and including the Base64 encoding of the binary contents of an example JPEG image. On the other hand, there is a simple XML-like condition specifying that the metadata field *FileSize* must be inferior to 1000 bytes. In order to deal with this model duality, MPQF operates over sequences of what the standard calls *evaluation-items*. By default, an *evaluation-item* (EI) is a multimedia content in the multimedia database, but other types

**Code 1** Example MPQF input query

---

```

<MpegQuery>
  <Query>
    <Input>
      <OutputDescription outputNameSpace="//purl.org/dc/elements/1.1/">
        <ReqField>title</ReqField>
        <ReqField>date</ReqField>
      </OutputDescription>
      <QueryCondition>
        <TargetMediaType>image/jpg</TargetMediaType>
        <Condition xsi:type="AND" preferenceValue="10">
          <Condition xsi:type="QueryByFreeText">
            <FreeText>Barcelona</FreeText>
          </Condition>
          <Condition xsi:type="GreaterThanOrEqual">
            <DateTimeField>date</DateTimeField>
            <DateValue>2008-01-15</DateValue>
          </Condition>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>

```

---

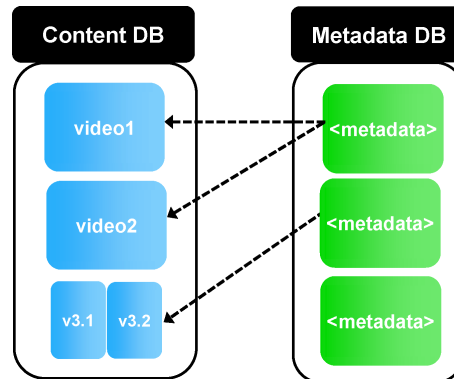


Fig. 3: Dual database model

of EIs are also possible. For instance, an EI can be a segment of a multimedia resource, or an XPath-item related to a metadata XML tree. The scope of query evaluation and the granularity of the result set (the granularity of EIs) can be determined by a *EvaluationPath* element specified within the query. If this *EvaluationPath* element is not specified, the output result is provided as a collection of multimedia contents, as stored in the repository, all satisfying the query condition.

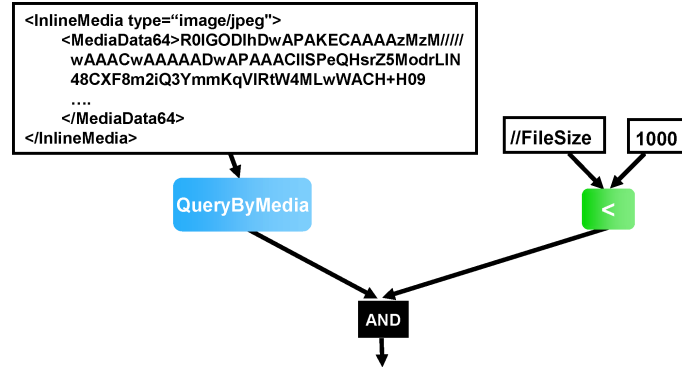


Fig. 4: Example condition tree

The *EvaluationPath* element determines the query scope on basis to a hypothetical XML metadata tree covering the entire database. So, we can redraw our visual representation of the model in order to show this hierarchical nature, as shown in Figure 5.

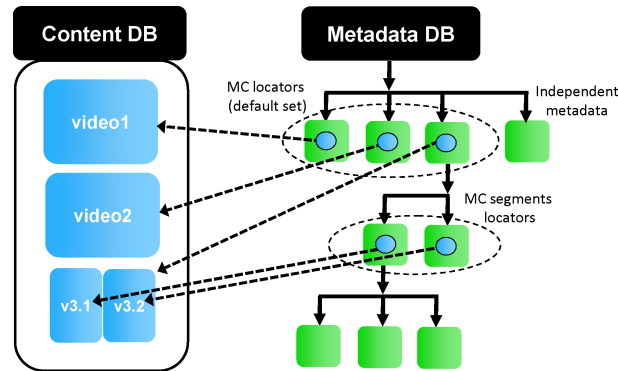


Fig. 5: Hierarchical MPQF model

### 2.3 MPQF evaluation model

The condition tree of an MPQF query is constructed combining filtering elements (conditions) from the *BooleanExpressionType* and interconnecting them with Boolean operators (*AND*, *OR*, *NOT* and *XOR*). Each condition acts over a sequence of evaluation-items and, for each one, return a value in the range of  $[0..1]$ . In the case of XML-like conditions (e.g. it "the size of the file must be smaller than 1000 bytes") the condition can return just 1 or 0, which mean *true* and *false* respectively. In the case of IR-like conditions (keywords or CBIR),

they can evaluate to any value in the range of  $[0..1]$ . A *threshold* value within a condition is used to indicate the minimum value the score of an evaluation-item is required to have. Otherwise the evaluation-item is not considered further during evaluation.

So, with respect to XML-like conditions, MPQF acts as a conventional Boolean-based filtering language, while with respect to IR-like conditions MPQF acts preserving scores as a fuzzy-logic system. The standard specifies the behaviour of the provided Boolean operators in presence of non-Boolean values.

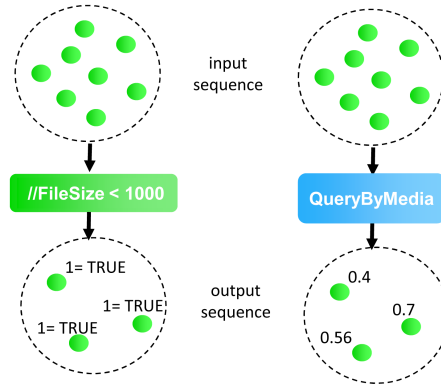


Fig. 6: Two evaluation styles in MPQF: Boolean and fuzzy-logic

### 3 ISO/IEC 24800 standard (JPSearch)

The selection of a unified query interface is not enough to guarantee interoperability if it is not accompanied with a proper mechanism to manage metadata heterogeneity. The need of dealing with the management of metadata translations is a common factor in all the approaches to distributed image search&retrieval. Currently there is a standard solution to this problem proposed by the JPEG Committee, named JPSearch (ISO/IEC 24800). JPSearch provides a set of standardized interfaces of an abstract image retrieval framework. On one hand, JPSearch specifies the pivot JPSearch's Core Metadata Schema as the main component of the metadata interoperability strategy in ISO/IEC 24800. The core schema contains a set of minimal core terms which serve as metadata basis supporting interoperability during search among multiple image retrieval systems. The core schema is used by clients to formulate, in combination with the MPEG Query Format, search requests to JPSearch compliant search systems. In addition to the definition of JPSearch Core Metadata Schema, ISO/IEC 24800 provides a mechanism which allows a JPSearch compliant system taking profit from proprietary or community-specific metadata

schemas. A translation rules language, JPSearch Translation Rules Declaration Language (JPTRDL), is defined, allowing the definition of translations between proprietary metadata schemas and the JPSearch Core Metadata Schema.

### 3.1 JPSearch Core Schema

The JPSearch Core schema defines the metadata structures which must be understood (directly or by means of applying translations to a native metadata format) by any JPSearch compliant system. The core schema has been developed by investigating available image metadata formats and selecting those elements or its semantic concepts that show a broad overlap (e.g. have an equivalent representation) among the individual metadata formats. A similar approach of choosing the core properties was also chosen by W3C group <sup>1</sup>. For instance, the element *Title* of our core schema has similar tag names in Dublin Core (*dc:title*), DIG35 (*ipr\_title*) or MPEG-7 (*\CreationInformation \Creation \Title*). Based on this analysis, the current version of the core schema supports 20 main elements including spatial annotation. The main elements represent basic information such as *Title*, *Description*, *Keyword*, *CreationDate* and so forth. The broader set concentrates on semantic descriptions of the content such as events, persons, etc. Furthermore, a mechanism for the integration of parts of other formats (e.g., low level descriptors of MPEG-7) during search is provided.

Code 2 shows an example XML image description compliant with the JPSearch Core Schema:

---

<sup>1</sup> [http://www.w3.org/2008/WebVideo/Annotations/drafts/ontology10/WD/mapping\\_table.html](http://www.w3.org/2008/WebVideo/Annotations/drafts/ontology10/WD/mapping_table.html)



**Code 2** Example JPSearch image description

---

```

<ImageDescription>
  <Identifier>urn:unique:identifier:1:2:3</Identifier>
  <Creators>
    <GivenName>John</GivenName>
    <FamilyName>Smith</FamilyName>
  </Creators>
  <CreationDate>2011-12-17T09:30:47.0Z</CreationDate>
  <ModifiedDate>2011-12-17T09:30:47.0Z</ModifiedDate>
  <Description>Sample description</Description>
  <Keyword>Sardinia</Keyword>
  <Keyword>Italy</Keyword>
  <Keyword>50th JPEG meeting</Keyword>
  <Title>Example Instance document of the JPSearch core schema</Title>
  <GPSPositioning latitude="34" longitude="34" altitude="10"/>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <Description>A short description about the selected region
    </Description>
    <Keyword>plenary meeting</Keyword>
  </RegionOfInterest>
  <Width>640</Width>
  <Height>480</Height>
</ImageDescription>

```

---

**3.2 JPSearch Translation Rules Declaration Language (JPTRDL)**

The JPSearch Translation Rules Declaration Language (JPTRDL) allows the publication of machine-readable translations between metadata terms belonging to proprietary metadata schemas and metadata terms in the JPSearch Core Metadata Schema. Users can choose which metadata language to use in a JPSearch-based interaction if the proper translations are available. JPTRDL is a key component of the metadata interoperability strategy in ISO/IEC 24800, and it aims to allow a JPSearch compliant system benefit from proprietary or community-specific metadata schemas.

Code 3 shows a one-to-many translation rule which maps the JPSearch Core Schema date element into three fields.

**Code 3** Example JPSearch translation rule

---

```

<?xml version="1.0" encoding="iso-8859-1"?>
<TranslationRules>
  <TranslationRule xsi:type="OneToManyFieldTranslationType">
    <FromField xsi:type="FilteredSourceFieldType">
      <XPathExpression>date</XPathExpression>
      <FilterWithRegExpr>(\d\d)/(\d\d)/(\d\d\d\d)</FilterWithRegExpr>
    </FromField>
    <ToField xsi:type="FormattedTargetFieldType">
      <XPathExpression>day</XPathExpression>
      <ReplaceWithRegExpr>£1</ReplaceWithRegExpr>
    </ToField>
    <ToField xsi:type="FormattedTargetFieldType">
      <XPathExpression>month</XPathExpression>
      <ReplaceWithRegExpr>£2</ReplaceWithRegExpr>
    </ToField>
    <ToField xsi:type="FormattedTargetFieldType">
      <XPathExpression>year</XPathExpression>
      <ReplaceWithRegExpr>£3</ReplaceWithRegExpr>
    </ToField>
  </TranslationRule>
</TranslationRules>

```

---

**3.3 JPSearch Registration Authority**

According to the JPSearch specification, ISO/IEC 24800 compliant systems can manage multiple proprietary or community-specific metadata schemas, besides the JPSearch Core Metadata Schema. The multiplicity of schemas is solved by allowing the publication of machine-readable translations between metadata terms belonging to proprietary metadata schemas and metadata terms in the JPSearch Core Metadata Schema. In order to rationalize the usage of schemas and translation rules across different JPSearch systems, Subclause 3.3.3 of Part 2 of ISO/IEC 24800-2 specifies that a global authority for schemas and their translation rules will be established where all JPSearch compliant retrieval applications can obtain the information needed.

The establishment of a JPSearch Registration Authority (JPSearch RA) was formally approved during the 54th JPEG meeting in Tokyo, Japan, in February 2011, and will be operative in July 2011. The JPSearch RA will maintain a list of Metadata Schemas together with their related Translation Rules, if any. Those schemas and rules will be directly stored in the JPSearch RA web site or the JPSearch RA web site will provide a link to an external organization in charge of keeping that information updated. Registration forms will be available from the Registration Authority. Any person or organization will be eligible to apply. More information about the JPSearch RA can be obtained at [www.iso.org/iso/maintenance\\_agencies/or](http://www.iso.org/iso/maintenance_agencies/or) directly at the JPEG home page ([www.jpeg.org](http://www.jpeg.org)).

## 4 Unifying MPEG Query Format and JPSearch, a proposal

IT professionals aiming to take profit from the MPQF and JPSearch standards will encounter several obstacles. In principle, the straightway approach would be to just apply the JPSearch specification, as it defines the complete framework and its Part 3 relies in MPQF as the query language. However, the way JPSearch adapts MPQF to the still images domain suffers from several flaws, which results in queries which are not complaint with MPQF. These defects can be summarized as:

- JPSearch Part 3 [4] redefines the XML root hierarchy of the MPQF queries (called JPQF queries in that context). Figure 7 shows graphically the XML root hierarchy of JPQF, which is different to the one in Figure 2. These differences make JPQF queries not compliant with the MPEG Query Format standard. This is inconvenient, especially considering that JPQF is just the subset of MPQF which does not deal with audio or video.
- Because JPSearch Part 3 replicates several syntax elements from MPQF instead of referencing them, it does not consider the latest corrigenda and amendments from MPEG. For instance, the 2nd MPQF corrigenda [7] fixes an error in the MPQF's *QueryConditionType* which disallowed the expression of empty queries (queries without conditions). Because *QueryConditionType* is redefined in JPSearch Part 3 the error has not been fixed there.

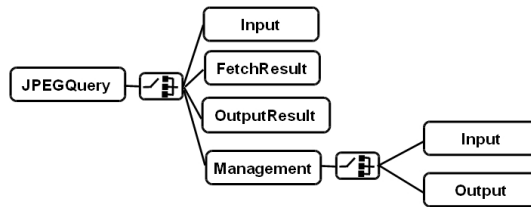


Fig. 7: JPSearch Part 3 (query format) root elements

In order to solve these problems we propose:

1. Withdraw JPSearch Part 3 and, optionally, simply refer to the MPEG Query Format as the query language for JPSearch compliant systems.
2. Within the MPEG Query Format standard, establish the JPSearch metadata strategy as the normative way to handle metadata in MPQF. MPQF is metadata neutral and does not specify neither predefined metadata structures nor visual search descriptor formats to which address the queries. This poses an interoperability problem cause queries referring to different metadata schemas or CBIR descriptors would be incompatible. Taking profit from the JPSearch metadata approach would provide MPQF with the metadata

infrastructure it currently lacks, but keeping it open to other metadata formats as the JPSearch metadata strategy also consider this possibility. This approach also would allow MPQF to take benefit from JPSearch Registration Authority (JPSearch RA).

Example in Code 4 shows an MPQF query addressing metadata fields as defined in the JPSearch Core Schema.

---

**Code 4** Example MPQF input query addressing metadata fields as defined in the JPSearch Core Schema

---

```

<MpegQuery>
  <Query>
    <Input>
      <OutputDescription outputNamespace="//purl.org/dc/elements/1.1/">
        <ReqField>JPCore:Identifier</ReqField>
        <ReqField>JPCore:Creators/JPCore:GivenName</ReqField>
        <ReqField>JPCore:Creators/JPCore:FamilyName</ReqField>
        <ReqField>JPCore:CreationDate</ReqField>
        <ReqField>JPCore:Width</mpqf:ReqField>
        <ReqField>JPCore:Height</mpqf:ReqField>
        <ReqField>JPCore:GPSPositioning/@latitude</ReqField>
        <ReqField>JPCore:GPSPositioning/@longitude</ReqField>
      </OutputDescription>
      <QueryCondition>
        <TargetMediaType>image/jpg</TargetMediaType>
        <Condition xsi:type="Equal">
          <DateTimeField>JPCore:CreationDate</DateTimeField>
          <DateTimeValue>2009-10-07T08:46:45</DateTimeValue>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>

```

---

## 5 MPEG Query Format visual search operations and limitations

Visual search Examples in Code 5 and Code 6 show two possible ways to specify a query-by-example condition in MPQF. *QueryByMedia* and *QueryByDescription* query types are the operations of MPQF which allow specifying information about an example multimedia content. The individual difference lies in the used sample data. The *QueryByMedia* query type uses a media sample such as image or video as a key for search, whereas *QueryByDescription* allows querying on the basis of an XML-based description key. The example query in Code 5 embeds an example JPEG image in Base64 encoding. It is also possible to just specify the URI of the example image.

---

**Code 5** Example MPQF input visual search query inlining the query image in Base64 encoding

---

```

<MpegQuery>
  <Query>
    <Input>
      <OutputDescription outputNamespace="//purl.org/oc/elements/1.1/">
        <ReqField>title</ReqField>
        <ReqField>date</ReqField>
      </OutputDescription>
      <QueryCondition>
        <TargetMediaType>image/jpg</TargetMediaType>
        <Condition xsi:type="QueryByMedia">
          <MediaResource xsi:type="MediaResourceType" resourceID="id1">
            <MediaResource>
              <InlineMedia type="image/jpeg">
                <MediaData64>R0lGODlhDwAPAKECAAAAzMzM////wAAACwAAAAADwA
                PAAACIIISPeQHsrZ5ModrLlN48CXF8m2iQ3YmmKqVlRtW4MLwWACH+H09
                wdGltaxplZCBieSBVbGVhZCBtbWYdFNhdmVyIQAAOw==</MediaData64>
              </InlineMedia>
            </MediaResource>
          </MediaResource>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>

```

---

The example query in Code 6 shows how an example description can be included in a query to form a condition. The query requests JPEG images which expose similar descriptors to the attached MPEG-7 metadata descriptor. Descriptors containing low-level features (e.g. the MPEG-7 *Color Layout* descriptor) can be used, becoming a way to express CBIR queries.

---

**Code 6** Example MPQF input visual search query referring to the query image as a URI

---

```
<?xml version="1.0" encoding="UTF-8"?>
<MpegQuery>
  <Query>
    <Input>
      <QueryCondition>
        <TargetMediaType>image/jpeg</TargetMediaType>
        <Condition xsi:type="QueryByDescription" matchType="exact">
          <DescriptionResource resourceID="desc07">
            <AnyDescription xmlns:mp7="urn:mpeg:mpeg7:schema:2004">
              <mp7:Mpeg7>
                <mp7:DescriptionUnit xsi:type="mp7:CreationInformationType">
                  <mp7:Creation>
                    <mp7:Title>World Cup 2006</mp7:Title>
                  </mp7:Creation>
                </mp7:DescriptionUnit>
              </mp7:Mpeg7>
            </AnyDescription>
          </DescriptionResource>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

---

The main limitation of the MPQF's operations for expressing CBIR queries is the lack of reference visual descriptors and search parameters. MPQF's only provides placeholders where any information about an example image can be introduced, without constraining which are the valid descriptor formats and which kind of visual query has to be performed. It's up to the server to decide how to process the example information. This feature clearly constraints the interoperability of visual search operations.

### 5.1 Enriching the MPEG Query Format with predefined compact descriptors for visual search

Recently MPEG has issued a Call for Proposals for compact descriptors for visual search (CDVS) [6]. This initiative aims to standardizing technologies for visual content matching in images or video, including the format of visual descriptors and descriptor extraction process. The future standard will ensure interoperability of visual search applications and databases as well as reduce load on wireless networks carrying visual search-related information. Without still knowing which would be the format of the visual descriptors in the future standard, it is clear that the MPEG Query Format should include them at least as the default way of expressing visual search conditions. The flexibility of the *QueryByDescription*

query type will allow to directly inline the descriptor within the query or to just refer to the descriptor URI.

## 6 Conclusions

In this paper we have evaluated the standards ISO/IEC 15938-12 (MPEG Query Format, MPQF) and ISO/IEC 24800 (JPSearch) as candidates to provide interoperability to distributed visual search query interfaces. MPQF has proven to be a powerful multimedia query language, while JPSearch provides the necessary mechanisms to address the problem of metadata interoperability. However, we have identified some problems related to the integration of both standards and to the way they cover the visual search scenario. On one hand, our proposed solution consists on including the JPSearch metadata interoperability model within the MPQF standard, while fixing the way the MPQF language is adapted within JPSearch's Part 3. On the other hand, we suggest to include the future MPEG standard for compact descriptors for visual search (CDVS) as the default way of expressing visual search conditions within MPQF. The MPEG Query Format standard was published in December 2008. After almost three years, two amendments and two corrigenda, we envisage that the time has come for a second edition of ISO/IEC 15938-12, which could include the improvements proposed in this paper.

**Acknowledgments.** This work has been partly supported by the Spanish government (TEC2008-06692-C02-01).

## References

1. Mario Döllner, Ruben Tous, Matthias Grubne, Kyoungro Yoon, Masanori Sano, and Ian S. Burnett. The MPEG Query Format: On the Way to Unify the Access to Multimedia Retrieval Systems. *IEEE Multimedia*, ISSN: 1070-986X, volume 15, issue 4, 2008.
2. Matthias Grubne, Ruben Tous, Mario Döllner, Jaime Delgado, and Harald Kosch. MP7QF: An MPEG-7 Query Format. In *Proceedings of the 3rd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS 2007)*, pages 15–18, Barcelona, Spain, November 2007.
3. ISO/IEC 15938-12:2008 Information Technology - Multimedia Content Description Interface - Part 12: Query Format, 2008.
4. ISO/IEC 24800-3:2010 Information technology - JPSearch - Part 3: JPSearch Query format, 2010.
5. ISO/IEC 24800-2:2011 Information technology - JPSearch - Part 2: Registration, identification and management of schema and ontology, 2011.
6. MPEG's Requirements Subgroup. ISO/IEC JTC1/SC29/WG11/N12038 Call for Proposals for Compact Descriptors for Visual Search. Output document of the 96th MPEG meeting, March 2011, Geneva, CH , 2011.
7. ISO/IEC 15938-12:2008/Cor.2:2010, Information technology – Multimedia content description interface – Part 12: Query format, TECHNICAL CORRIGENDUM 2 , 2010.