

用 MPEG-7 DDL 表示用户查询需求的方法^{*})

富 亮 薛向阳

(复旦大学计算机科学与工程系 上海200433)

摘 要 随着 ISO MPEG-7 标准在多媒体信息检索领域的深入应用,如何描述用户的查询信息需求变得越来越重要,本文提出一种新的用户查询需求的表示方法。该方法的主要特点是充分考虑 MPEG-7 标准对多媒体内容的标准化描述特性,因此对用户提交的信息需求具有很强的表达能力和较好的通用性。

关键词 MPEG-7, DDL, 描述方案(DS), 描述子(D), 信息检索, 查询语言

Using MPEG-7 DDL to Represent Users' Query Needs

FU Liang XUE Xiang-Yang

(Dept. of Computer Science and Engineering, Fudan University, Shanghai 200433)

Abstract By the wide development of ISO MPEG-7 standard in multimedia information retrieval, it becomes even more important to describe users' information query needs. This paper proposes a new method to represent query information needs based on MPEG-7 standard in order to apply on video query. The mainly features of this method is that it considers the multimedia content description standard of MPEG-7 thoroughly, so it has very strong expression ability and good universality to formalize users' information query needs.

Keywords MPEG-7, DDL, Description scheme, Descriptor, Information retrieval, Query language

1 引言

近年来,针对多媒体信息各种层出不穷的应用, MPEG 组织制定出一个新的标准 MPEG-7^[1], 它为各种媒体的检索应用定义了描述规范。在 MPEG-7 定义的描述框架中, 包含了描述子(Ds)、描述方案(DS)和描述定义语言(DDL), 多种多样的媒体信息将在此描述框架下得到标准化和形式化的描述, 一旦实现了对多媒体信息内容的形式描述和表示, 则多媒体信息检索和过滤等应用将会得到迅速普及。本文主要目的是以视频信息检索为例, 面向标准化的视频内容的描述数据, 设计一个能充分和精确表达用户查询需求的形式化的表示方法。

在可视化查询语言的研究领域, 过去曾提出过多种查询语言来表示用户的查询需求, 大部分都是基于类似 SQL 或 QBE 的可视化查询语言^[2]。近年来, 比较著名的查询语言是 VISUAL^[3], 这是一种系统的、非过程性的、面向对象的查询语言, 它使用了样例查询(Query-by-Example)方式来提交查询对象。SVIQUERL^[4]则利用可视化信息搜寻(VIS: Visual Information Seeking)概念把查询向二维空间数据上扩展; VQL^[5]则是在基于关系模型的基础上提出的一种面向对象的查询语言; QBD^[6]提供了一种基于图表的查询方式。显然, 这些研究都是采用设计查询语言的思想来表示用户的查询需求。

在 XML 数据库研究领域, 也出现了一些针对 XML 文档的查询语言来表示用户的查询需求, XIRQL^[7]就是一种从 XML 文档中提取信息的语言, 它主要是整合了信息检索中概

率模型以及权重、排序、相关反馈等技术, 由于它适合于通用的非特定领域的 XML 文档的检索, 因此并不能很好处理特定领域的 XML 文档, 例如 MPEG-7 中定义的各种描述多媒体数据的 XML 文档。W3C 曾提出过一种 XML-QL^[8]的查询语言, 它结合了 XML 和 SQL 特点, 是一种两者兼顾的语言。今天, 考虑到 XML 技术日臻完善, 如果可以用 XML 本身的表示能力来描述用户的查询信息需求, 可能会更加便利。XML-GL^[9]就是一种为了适应 Web 的半结构化特点而设计的查询语言, 充分考虑了 XML 本身的特点, 但是它主要针对文本信息的 XML 检索, 对多媒体信息的查询来说还存在欠缺。

在多媒体数据库查询领域, 也提出过不少查询语言以满足用户的查询需求, 但它们大都是类 SQL 的。如 VIQS^[10]是根据视频数据的特点对查询条件作了扩展; MOQL^[11]是一种比较通用的、由 OQL 发展成的多媒体查询语言, 能够支持多媒体数据在时间上和空间上的关系。还有一种较为早期的 VideoSQL^[12]是专门针对视频数据库的查询语言。但是这些用于多媒体数据库的查询语言对于目前广泛被采用的 MPEG-7 描述数据而言, 并不能提供最好的支持。

随着 MPEG-7 得到越来越普遍的应用, 尤其在视频信息检索领域, 用户的需求也随之变得非常复杂, 例如:

1. 找出所有足球比赛进球的镜头并且根据不同的进球球员分组。
2. 找出和用户给出的图像相似的场景并且要求这个场景中某位名人。

这些需求包含了对查询结果在时间和空间上的各种复杂

^{*})本文工作受到国家自然科学基金(60003017, 60373020, 69935010)、八六三计划(2001AA114120, 2002AA103065)以及上海市科技发展基金(03DZ15019, 03DZ14015)共同资助。富 亮 硕士研究生, 主要从事视频信息检索研究。薛向阳 教授, 博导, 主要从事多媒体信息检索与处理等研究。

关系的组合,同时这些需求在与多媒体数据库检索时也存在各种相互之间的复杂关系。由此可见,主要需要解决以下两个问题:

1. 需求表示如何能充分反映用户的意愿,以适应各种复杂时空关系的用户需求,以得到最佳的检索结果。

2. 需求表示如何能充分结合 MPEG-7 的多媒体信息描述,以达到和数据源的最佳耦合关系,以实现最有效的检索。

本文旨在采用基于 MPEG-7 标准中的描述定义语言(DDL),以视频数据库为例,来实现一种针对 MPEG-7 多媒体信息检索应用的、可扩展的、能够表示各种复杂查询需求的方法。而且这样的查询表达也具有清晰、简单、通用和灵活的特点。其可行性基于以下几点:

1. MPEG-7 标准在视频等多媒体信息检索领域的广泛应用,它所定义的多媒体数据模型表达能力非常强大,具有普遍适应性。

2. MPEG-7 的描述是基于 XML 的,因此用 DDL 描述的查询需求有良好的可扩展性,可以非常方便地扩展到各个多媒体应用领域。

3. 如果多媒体信息库与用户查询需求都采用了同样的描述体系,那么就能使多媒体检索系统有了更好的一致性和完整性。

2 研究背景

2.1 多媒体信息检索系统的通用框架

图1给出了多媒体信息检索系统的通用框架,下面分别介绍各个组成模块的功能。

(1)多模态用户界面 该模块负责整个检索系统的输入和输出,实现与用户的交互。用户需要通过该模块提出查询需求。对于多媒体信息检索来说,这是一个多模态的输入过程,因为用户可能会提交一段文本、一幅或多幅样例图像、一段或多段样例视频片段、一段话音等来表达其查询意愿。用户再通过界面得到检索系统的返回结果。观看和聆听查询结果同样是一个多模态输出过程,因为用户可能会获得关于多媒体数据的描述信息、一系列相关的图像、一段相关的语音或一组相关的镜头等。由此可见,不管是输入信息还是输出内容,都要求用户通过多模态的人机界面提交或定义,它是多媒体信息检索系统中必不可少的非常关键的组成部分。

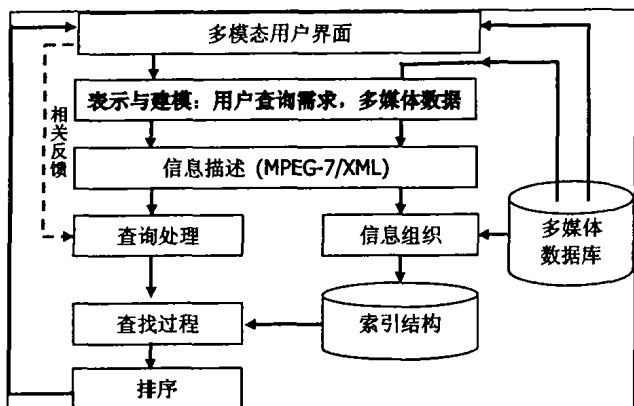


图1 多媒体信息检索系统的通用框架

(2)表示和建模 用户输入的多模态性决定了查询信息的多样性和复杂性,因此,如果缺少对用户查询需求的表示和建模,就无法清晰和准确地表示用户的真正查询意愿,也就很难实现后续的查询处理模块。对于多媒体数据库中存储的多

媒体信息同样需要进行建模和表示,只有建立了统一的模型,才能够满足各种查询需求。由此可见,对用户提交的查询需求信息的建模和表示,实际上和对数据库中多媒体数据的建模和表示是一致的。不同之处仅仅在于:前者是在线处理,后者是离线处理。

(3)信息描述 该模块是对用户查询需求和多媒体数据的建模所得到的中间数据进行形式化描述,例如用 MPEG-7 标准中定义的各种类型的描述子和描述方案来刻画多媒体数据的内容,类似地,必须研究如何用 MPEG-7 描述子和描述方案来表示用户的查询需求。因此,无论是数据库中的多媒体数据,还是用户提交的查询信息需求,通过该模块的处理都会得到统一的形式化表示。

(4)信息组织 该模块根据所采用的描述子和描述方案的数据类型,采用相应的索引结构重新组织描述数据,以实现高效的相似检索。例如,对于文本类型的描述子可采用倒排文件组织;对于高维矢量数据可能要采用 VA-File。

(5)查询处理 该模块根据用户查询需求的形式化描述,进一步为后续查找过程的实现作再处理,以便与多媒体数据的索引结构相匹配。另外,它还要处理用户的相关反馈信息,根据反馈信息,调整查询算法,以期取得更加符合用户意愿的查询结果。

(6)查找过程 该模块采用各种相似匹配算法,在索引结构之上快速查找相关的多媒体信息。

(7)排序 该模块把查找过程中找到的相关结果,根据相似度或相关度大小进行排序。

2.2 多媒体内容描述的规范化标准

基于多媒体信息检索的一般要求,MPEG 组织提出了 MPEG-7 标准。在该标准的框架中包含了描述子(Ds)、描述方案(DSs)、描述定义语言(DDL)以及编码方案。它们分属于不同的工作小组^[16]。例如 MDS (Multimedia Description Scheme)小组就是1999年7月正式建立的一个工作组,其任务是开发多媒体描述方案,定义 MPEG-7 标准中的第五部分“多媒体内容描述接口”^[17]。其包括基本元素,内容描述,内容管理,内容组织,浏览与访问以及用户交互六个部分。

内容描述工具是从多媒体数据的结构和语义方面来描述可感知的多媒体数据的内容。其中,结构描述工具刻画了多媒体数据在空间和时间上的结构,其基本分割单位是片段(Segment),通常把片段分为静止区域、视频片段和音频片段三部分。它们分别刻画了片段的属性、子片段分解以及片段与片段间的关系。语义描述工具刻画的是多媒体数据的现实含义,它包括的语义实体有对象、事件、概念等。同样,它也刻画了多个语义实体间的关系。

因此,一段视频信息即由结构化描述和语义化描述组成。对于结构化描述,MPEG-7 提出了一个描述方案是对可编辑的视频片段的剪辑分析(AnalyticEditedVideoSegment),它可以描述基于一定内容的具有分析意义的视频片段。例如,可以描述一组镜头和镜头间跳变组成的视频片段。下面的文档描述的是一个镜头的部分信息和一个跳变的部分信息:

```
<VideoEditingDecomposition gap="false" overlap="false"
  <AnalyticClip id="C1" clipEditingLevel="shot"
    <MediaTime>...</MediaTime>
    <!-- some feature descriptions and relationship -->
  </AnalyticClip>
  <AnalyticTransition id="T1" transitionEditingLevel="
    global"
    <MediaTime>...</MediaTime>
    <EvolutionType><Name>crossdissolve</Name></Evolu-
    tionType>
```

```

</AnalyticTransition>
<AnalyticClip id="C2" ...
</VideoEditingDecomposition>

```

对于语义化描述, MPEG-7也提供了多种描述方案, 包括各种语义上的实体: 对象、事件、代理对象、概念、语义状态、语义地点、时间以及它们的属性和它们之间复杂的相互关系。对于视频信息, 可以自动或人工给出其中的对象、事件和关系等。例如以下描述了一段足球比赛的视频片段的部分语义, 其标签信息是“forward player”, 并对此做了详细语义说明:

```

<AgentObject id="AO1">
  <Label><Name> Forward player</Name></Label>
  <Agent> <Name> <FamilyName> Morientes </FamilyName> </
  Name> </Agent>
  <Definition> <FreeTextAnnotation>
    Forward player for the Spanish soccer team whose name is
    Morientes
  </FreeTextAnnotation> </Definition>
  <MediaOccurrence> <MediaLocator>
    <MediaUri> http://www.mpeg7.org/game.mpg </Me-
    diaUri>
    <MediaTime>... </MediaTime>
  </MediaLocator> </MediaOccurrence>
</AgentObject>

```

由此可见, 目前 MPEG-7标准已经规定了多媒体信息各种描述形式。但为了能够快速、有效地实现多媒体信息检索任务, 需要提出合理的描述方法来清晰准确地定义用户的查询需求, 然而在这一点上, MPEG-7标准没有做出详细规定, 即还没有对用户查询需求进行规范化描述。

3 用户查询的需求表示

3.1 一个查询需求的例子

为了说明查询需求的表示方法, 这里以视频查询为例先给出一个具体的用户需求:

用户希望得到与“足球比赛”这一事件相关的100个视频场景以及这些场景的视频格式、长度等媒体信息。这些场景来自前一查询的返回(如 ESPN 体育台), 并要出现与用户提交的图像 A(如某足球场)相似的帧, 还要包含用户提交的一个视频片段 B(如某球星射门镜头)。最后, 返回结果要根据不同的媒体格式分组, 每一组中再根据相似度由大至小降序排序。

对于这样复杂的用户需求, 我们可以采用如下的表示方法:

```

<VideoQuery id="Q1">
  <QueryInput id="Q1-A" mode="FrameMode">
    <Location type="URL">http://...</Location> <!--图像 A-->
  </QueryInput>
  <QueryInput id="Q1-B" mode="ClipMode">
    <Location type="File">c:\...</Location> <!--视频 B-->
  </QueryInput>
  <QueryOutput amount="100"> <!--返回数-->
    <VideoObject id="Q1-O0">Scene</VideoObject> <!--返回视频
    类型-->
    <MediaInformation id="Q1-O1">MediaProfile </MediaInfor-
    mation> <!--返回媒体信息-->
  </QueryOutput>
  <QueryRange>Video</QueryRange>
  <QueryConstraint>
    <TextCondition>
      <CreationCondition id="Q1-C0">
        <Creation><Creator>
          <OrganizationRef ref="#Q0-O1"><OrganizationRef>
            <!--相关组织引用(Q0查询略)-->
          </OrganizationRef>
        </Creator><Creation>
      </CreationCondition>
      <SemanticCondition id="Q1-C1">
        <Event><Label><Name>足球比赛</Name></Label></
        Event> <!--相关事件-->
      </SemanticCondition>
    </TextCondition>
    <MMCondition id="Q1-C2" name="before" source="#Q1-A"
    target="#Q1-B"/><!--具体条件-->
  </QueryConstraint>
  <QueryRelation id="Q1-R0" name="AND" source="#Q1-C0"

```

```

target="#Q1-C1"/> <!--条件组合-->
  <QueryRelation id="Q1-R1" name="AND" source="#Q1-R0"
  target="#Q1-C2"/>
  <GroupCondition>MediaFormat</GroupCondition> <!--结果分组-->
  <SortCondition direction="desc"> <!--结果排序-->
    <Similarity>Video</Similarity>
  </SortCondition>
</QueryConstraint>
</VideoQuery>

```

从描述中可以看到, 一段复杂的用户需求被清晰地用基于 MPEG-7 的描述方法表述出来。这样的表述不仅有利于对用户意愿的表达, 而且也有利于与 MPEG-7描述的数据源的相互关系。这将在下文中作出详细的阐述。

3.2 查询需求的基本元素

在 MPEG-7对多媒体信息的描述标准的定义下, 就需要提出对查询需求的描述。一般而言, 查询语言无论其复杂度如何, 它们都有需要满足的基本要求^[9], 同样从前文的示例中也可以看出, 对多媒体信息查询需求的描述也应有如下的基本元素:

(1) 查询输入 确定了查询输入的方式和内容, 对于多媒体数据的查询, 除了最基本的文本关键字查询方式之外, 需要提供通过多媒体数据的方式来查询多媒体数据的能力, 即样例查询。例如, 在前文的示例中, 用户提交一个视频片段 B 作为查询样本, 希望在视频数据库中寻找含有它的场景, 或者提交图像在图像库中寻找相似的纹理, 提交音频在音频库中寻找说话人等。因此, 查询样例可以是图像、镜头、场景、音频等各种可能多媒体模态的输入及其组合。

(2) 查询输出 确定了查询输出的方式和内容, 对于多媒体数据的查询, 除了如前文示例中输出的媒体信息等文本信息之外, 最主要还需输出各种可能的多媒体模态, 如图像、镜头、场景等各种组合。

(3) 查询范围 确定了查询需求所关心的范围。对于多媒体信息检索, 可以根据不同的多媒体应用领域, 划分为视频、图像、音频等。如果仅在视频信息检索中, 也可以由多种范围划分的方法, 如地域、语言、来源等。例如在所有 BBC 新闻视频中检索, 或从所有电影片段中检索等。

(4) 查询约束 确定了查询需求所需要满足的各种约束条件和相互关系。一般而言, 这种约束总体上可以看作是由用户需要查询的文本查询条件和多媒体查询条件所组成的布尔复合关系和分组排序条件。文本查询条件包括数据源的属性和特征, 比如含有某个人物或地方的属性如前文示例中的“足球比赛”这一事件等。多媒体查询条件包括用户需要查询的多媒体对象与用户提交的查询样例之间的各种关系如前文示例中就要求与图像 A 相似并且还含有片段 B, 或者要求在左上方出现与查询样例相似的图像, 或查询同时含有两个说话人的音频等等。分组排序条件则是对整个查询结果根据某个属性或特征进行分组和排序如前文中根据“媒体格式”分组并按相似度排序等。

3.3 查询需求的 DDL 描述

多媒体信息查询的复杂性使得人们不可能完全按照传统查询语言的方式来实现查询, 因此我们采用与 MPEG-7标准一致的 DDL 描述方式。查询需求与数据源描述的这种统一, 更有利于后续的具体查询处理, 这一点将在后文中作详细阐述。限于篇幅, 这里只给出关于视频信息查询的需求表示, 其它媒体的需求表示在此描述基础上是很容易扩展的。

(1) 视频查询类型(VideoQueryType) 这一类型是整个视频信息需求描述方法的类型, 它包含了以下的查询输入类

型,查询输出类型,查询范围类型和查询约束类型。

VideoQueryType 的 DDL 描述如下:

```
<element name="VideoQuery" type="VideoQueryType"/>
<complexType name="VideoQueryType">
  <sequence>
    <element name="QueryInput" type="QueryInputType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="QueryOutput" type="QueryOutputType"/>
    <element name="QueryRange" type="QueryRangeType"
      minOccurs="0"/>
    <element name="QueryConstraint" type="QueryConstraintType"
      minOccurs="0"/>
  </sequence>
  <attribute name="id" type="mpeg7:ID" use="required"/>
</complexType>
```

(2) 查询输入类型(QueryInputType) 在这一类型中,如前所述有下列查询输入方式:文本关键字、图像、视频片段、音频片段等。对于文本查询方式,用户的查询需求在查询约束中给出,因此无需提交额外查询输入;对于图像等多媒体查询方式,用户需要把该图像、视频片段或音频片段提交出来,提交的方法可以是:以 URL 的形式提交,以文件的形式提交,以 BASE64 码流的形式提交,或者可能是相关反馈时视频库中的标识 ID 提交等。所以查询需求需要考虑这些不同的查询输入方式。并且多种查询输入方式具有可组合性。

QueryInputType 的 DDL 描述如下:

```
<complexType name="QueryInputType">
  <sequence>
    <element name="Location" type="string">
      <complexType>
        <simpleContent>
          <extension base="string">
            <attribute name="type" use="required">
              <simpleType>
                <restriction base="string">
                  <enumeration value="File"/>
                  <enumeration value="URL"/>
                  <enumeration value="UID"/>
                  <enumeration value="BASE64"/>
                </restriction>
              </simpleType>
            </attribute>
          </extension>
        </simpleContent>
      </complexType>
    </element>
  </sequence>
  <attribute name="id" type="mpeg7:ID" use="required"/>
  <attribute name="mode" use="required">
    <simpleType>
      <restriction base="string">
        <enumeration value="FrameMode"/>
        <enumeration value="ClipMode"/>
        <enumeration value="AudioMode"/>
      </restriction>
    </simpleType>
  </attribute>
</complexType>
```

查询输入样例可以有多个,这里需要指出的是,每一个样例都对应于一个 id,它主要用来标识一个图像帧(Frame)或视频片段(Clip)或音频片段(Audio)。它们将在查询条件中被指定布尔关系。如果没有查询样例则表示仅使用关键字文本查询的方式。

(3) 查询输出类型(QueryOutputType) 对于视频信息的检索,查询输出对象可以是图像帧、镜头、镜头组、场景和视频片段,但同时用户可能需要额外获得与视频信息相关的文本信息,这包括语义信息,如人物、地点、事件,还包括媒体信息,如片长、来源、尺寸等。同样,对于文本信息和视频信息的组合输出要求也是允许的。同时这一类型还包括一些如返回数量等其它查询输出要求。

QueryOutputType 的 DDL 描述如下:

```
<complexType name="QueryOutputType">
  <sequence>
    <element name="VideoObject" type="VideoObjectType"/>
    <element name="MediaInformation" minOccurs="0" maxOccurs="unbounded">
      <simpleType>
        <restriction base="string">
          <enumeration value="MediaFormat"/>
          <enumeration value="MediaProfile"/>
          ...
        </restriction>
      </simpleType>
    </element>
    <element name="SemanticInformation" minOccurs="0" max-
```

```
Occurs="unbounded">
  <simpleType>
    <restriction base="string">
      <enumeration value="Agent"/>
      <enumeration value="Event"/>
      ...
    </restriction>
  </simpleType>
</element>
</sequence>
<attribute name="amount" type="Integer" use="required"/>
</complexType>
<simpleType name="VideoObjectType">
  <restriction base="string">
    <enumeration value="StillRegion"/>
    <enumeration value="Shot"/>
    <enumeration value="ShotGroup"/>
    <enumeration value="Scene"/>
    <enumeration value="VideoClip"/>
  </restriction>
</simpleType>
```

为了有利于与 MPEG-7 所定义的视频数据模型之间的查询,查询输出中的媒体信息,语义信息等相关内容,使用了 MPEG-7 中的描述规范^[17]来定义。即这里的各类输出的文本信息的名称有选择地使用 MPEG-7 的 MDS 中定义的描述方案(DS)和描述子(D)的名称,如:MediaFormat D, Agent DS, Event DS 等。此外,属性“amount”表示返回数量。

(4) 查询范围类型(QueryRangeType) 在这里基于 MPEG-7 的查询视频信息的需求中,我们把查询范围广义地定为一种:视频信息库。为了提高该方法的可扩展性,可以进一步定义范围,也可以把查询范围扩大到音频库、静态图像库、3D 物体库等。

QueryRangeType 的 DDL 描述如下:

```
<simpleType name="QueryRangeType">
  <restriction base="string">
    <enumeration value="Video"/>
  </restriction>
</simpleType>
```

(5) 查询约束类型(QueryConstraintType) 查询约束由文本信息相关和多媒体内容相似两部分相互组成,其中,文本信息相关可以是媒体信息的相关,也可以是语义信息的相关,因此这种相关匹配涉及的是自由文本的检索过程,这里就不赘述了。而对于多媒体内容相似则涉及与查询输入方式提交的多媒体数据之间的相互时空关系以及相似性计算。

此外,查询约束还包括查询结果的分组和排序条件。对于分组,主要是对上述所说的文本信息的分组,例如视频来源、电影风格等。而对于排序,是对检索结果根据某些文本信息或直接根据描述子之间的相似度进行排序。

QueryConstraintType 的 DDL 描述如下:

```
<complexType name="QueryConstraintType">
  <sequence>
    <element name="TextCondition" type="TextConditionType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="MMCondition" type="MMConditionType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="QueryRelation" type="QueryRelationType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="GroupCondition" type="GroupConditionType"
      minOccurs="0"/>
    <element name="SortCondition" type="SortConditionType"
      minOccurs="0"/>
  </sequence>
</complexType>
<complexType name="TextConditionType">
  <choice>
    <element name="MediaCondition" type="mpeg7:MediaInformationType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="CreationCondition" type="mpeg7:CreationInformationType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="UsageCondition" type="mpeg7:UsageInformationType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="SemanticCondition" type="mpeg7:SemanticType"
```

```

        minOccurs="0" maxOccurs="unbounded"/>
    </choice>
</complexType>
<complexType name="MMConditionType">
    <complexContent>
        <extension base="mpeg7:SegmentRelationType"/>
    </complexContent>
</complexType>
<complexType name="QueryRelationType">
    <complexContent>
        <restriction base="mpeg7:RelationType">
            <attribute name="name" type="string">
                <simpleType><restriction base="string">
                    <enumeration value="AND"/>
                    <enumeration value="OR"/>
                    <enumeration value="NOT"/>
                </restriction></simpleType>
            </attribute>
        </restriction>
    </complexContent>
</complexType>

```

查询约束条件的 DDL 描述主要由两部分组成:条件项和条件关系。条件项如前所述主要有文本查询条件和多媒体查询条件。其中,文本查询条件是根据 MPEG-7 的标准定义规范给出媒体信息或语义信息所需要符合的条件,它可以是关于媒体方面的条件,创建方面的条件,使用方面的条件以及语义方面的条件等。如查询需求是要得到某位球星射门的视频信息,这就牵涉到语义条件中“球星”这一代理人物条件和“射门”这一事件条件等。多媒体查询条件则是根据 MPEG-7 的标准中描述片段关系的描述方案(DS)给出查询输入样例之间在时空上的相关关系,如查询需求中要求查询的视频信息必须满足:与用户提交的 Frame1 相似的视频帧要在与 Clip2 相似的视频片段之前出现等。在多媒体查询条件中还可以进而约束相似性计算时的各描述子权重等(限于篇幅未作定义)。此外,这些条件如果来自于另一查询(如子查询)的结果,则只需给出对另一查询中的 id 的引用,如前文例子中的组织名称即是对前一查询结果的引用。最后,条件关系则给出以上各条件项之间的布尔组合关系。

GroupConditionType 的 DDL 描述如下:

```

<complexType name="GroupConditionType">
    <choice>
        <element name="MediaInformation">
            <simpleType><restriction base="string">
                <enumeration value="MediaFormat"/>
                <enumeration value="MediaIdentification"/>
                ...
            </restriction></simpleType>
        </element>
        <element name="SemanticInformation">
            <simpleType><restriction base="string">
                <enumeration value="Agent"/>
                <enumeration value="Event"/>
                ...
            </restriction></simpleType>
        </element>
    </choice>
</complexType>

```

分组条件主要就是根据某个媒体信息或者语义信息来分组的。这里的分组名称有选择地使用 MPEG-7 中的 MDS 相应的描述子(D)的名称,如 MediaFormat D, MediaIdentification D。

SortConditionType 的 DDL 描述如下:

```

<complexType name="SortConditionType">
    <sequence>
        <choice>
            <element name="Information" type="GroupConditionType"/>
            <element name="Similarity" default="Video"/>
        </choice>
        <element name="SortCondition" type="SortConditionType"/>
    </sequence>
    <attribute name="direction" default="asc">
        <simpleType><restriction base="string">
            <enumeration value="asc"/>

```

```

        <enumeration value="desc"/>
    </restriction></simpleType>
</attribute>
</complexType>

```

排序条件中根据文本信息排序的表示类似于 GroupConditionType,此外也可以根据返回的视频片段的相似度排序。同时,考虑到排序会有子排序次序,所以这一类型是可以嵌套的。

4 查询需求的表示与 MPEG-7 数据源的相互关系

查询需求的表示是整个检索系统的第一步,但也是极其重要的一步。查询需求的准确性和高效性直接影响到后续的检索工作乃至用户最后得到的查询结果。因此,根据信息检索框架的描述,在准确性的基础上为了能够提高检索的效率,需要经过查询处理把需求进一步转成系统的内部表示(如压缩编码),这样可以更有效地与同样以压缩编码以及高维索引结构表示的 MPEG-7 描述的数据源进行匹配。为了便于阐述,仍以视频数据源为例来描述查询需求在后续检索过程中的处理和相互关系。

4.1 查询输入的再处理

在查询输入的描述中给出了用户提交的多媒体数据,它们被检索系统获得后,需要进行再处理,例如对于图像 A,我们需要提取它的特征,得到各种描述子(Descriptors),这一过程与数据库中的 MPEG-7 的各种颜色、纹理等描述子提取是一样的。它是后续查询匹配的前提工作。

4.2 查询范围的确定

查询范围决定了以下匹配检索工作中的被检索对象的范围。如前文示例中的查询范围是 Video,因此,以下的匹配工作就是在视频库的范围中进行的。

4.3 查询的匹配检索

这是整个检索过程的核心。它的匹配过程根据用户需求中的约束条件来完成,例如前文示例中所描述的文本查询条件(TextCondition),它含有一个创建条件(CreationCondition),是关于创建组织的,这个描述继承了 MPEG-7 的 Creation DS,此外它还含有一个关于事件描述的语义条件(SemanticCondition),这个描述继承了 MPEG-7 中 Event DS。因此,在检索中可以在使用了 MPEG-7 来描述的数据源中方便地找到符合这些语义条件的视频数据描述。对于 XML 文档(或者实际的二进制压缩描述文档),在同样的 DTD/Schema 下这样的匹配工作会变得更易实现。

同样,对于多媒体查询条件(MMCondition),这里给出的是一个用户提交的图像 A 与视频片段 B 之间的时间关系,需要满足 A 的出现在 B 之前。这里的描述方法继承了 MPEG-7 中的 SegmentRelation DS。我们之所以采用了 MPEG-7 中的这个关系描述方案,是为了使得整个检索系统对于关系的解析和理解能够有一个统一的模型。这样,虽然我们可能不能直接从描述数据库中找到这样的空间或时间关系,但是通过前面查询输入的再处理时得到的描述子等特征,可以方便地与数据库中已存在的描述特征进行匹配,具体可以通过一些用于高维索引的检索模型来完成描述特征的匹配,最终得到符合这一时空关系的相应结果。

最后根据 QueryRelation 条件的布尔关系组合,如在示例中表明是两个“逻辑与”关系,是把先前的匹配结果加以组合。同时,考虑到由于多媒体查询条件的匹配要比文本查询条件匹配慢得多,因此我们可以在检索到的所有符合文本查询

条件的集合中在检索出符合多媒体查询条件的集合,这样做可以大大提高检索性能。为此,我们可以考虑在 QueryRelation 条件中先判断组合的优先次序关系,这样更有利于匹配检索的高效性。

至此,我们得到了符合用户提出的约束条件后的集合。在此基础上,再根据用户的分组、排序条件,最终把该集合进行分类和排序。从 DDL 描述中也可以看到,分组和排序的名称都是 MPEG-7 中定义的描述子(D)的名字,如前文的 MediaFormat,我们可以根据这个描述子来对集合分类,并根据相似度排序后得到输出集合。

整个匹配检索核心过程一般可以用以下的步骤表示:

1. 分析布尔组合关系,确定各种条件的个数和逻辑组合关系,按关系的组合先后排序,并尽可能把文本查询条件置于多媒体查询条件之前。

2. 按顺序从组合关系中读出一个条件;如果是多媒体查询条件,转4;如果读完,转6。

3. 在已确定的数据源范围内匹配这个文本查询条件,(如果是逻辑与,则同时缩小数据源范围)转2。

4. 在多媒体查询条件中可获得用户多媒体高维描述特征,在当前数据源范围中匹配高维特征。

5. 根据多媒体查询条件中的时空关系,在先前的匹配结果中寻找符合该关系的数据源,(如果该多媒体查询条件在整个布尔组合关系中是逻辑与,则同时缩小数据源范围)转2。

6. 读出一个分组条件;如果没有,转8。

7. 在当前的数据源范围中寻找符合分组条件名称,进行数据源分类,转6。

8. 读出一个排序条件;如果没有,转10。

9. 在每一分类中的数据源中,分别寻找符合排序条件的名称,进行数据排序,转8。

10. 得到最后符合用户约束条件的数据源。

4.4 查询的输出处理

根据前面的匹配检索工作所得到的符合用户需求的视频描述是非常庞大的,因为它可能包括整体视频的描述信息,各个场景、镜头的描述信息,并且每段描述信息又包括结构信息、语义信息等。然而对于用户来说他只关心他所需要的,因此,只要根据查询输出的要求,如前文示例中需要输出视频场景以及相关媒体信息,因此只需从这些描述中提取出相应的部分即可。从 DDL 描述中也可以看到,查询输出的需求名称都是 MPEG-7 中定义的描述方案(DS)的名字如前文的 MediaProfile,因此可以很方便地从结果数据中提取相应的 DS 信息返回给用户。

事实上,检索系统本身应完成下列任务:解释上述查询需求,形成系统内部的数据格式,提交给系统内检索引擎,进行相似匹配和检索和返回符合用户意愿的结果。因此,一旦用户的查询需求被表达成前文的需求文档之后,视频检索系统本身就可以根据上述需求,快速有效地自动去处理与数据源描述之间的各种复杂关系来完成检索匹配任务,并返回检索结果。

对于视频信息检索系统的开发人员来说,如果能熟悉上述基于 MPEG-7 的描述方法,就可以针对不同应用目的设计出不同的应用需求。在我们开发的多媒体信息检索系统中,已经成功应用本文所介绍的查询需求的表示方法。

结论 本文主要讨论了如何基于 MPEG-7 来描述用户的

查询需求,它充分考虑到 MPEG-7 的特点,尽可能利用其描述方案来设计描述格式,所以能满足基本的、通用的查询需求。

当然,本文的描述还不是一种完善的查询语言,仅是以视频数据库为例,给出了对用户查询需求的一种表示方法。限于篇幅,也没有给出其它功能;如统计函数功能、联合查询功能以及如具体的相似度权重要求等。但是,由于这样的表示方法是具有很强扩展性的,并且与 MPEG-7 数据源之间的关系也能很好地保持一致性,因此,只要我们稍加改进,就可以把它应用到各个不同的媒体以及特定的领域。

参考文献

- 1 MPEG. The MPEG Home Page. <http://mpeg.telecomitalia.com/>
- 2 Batini C, et al. Visual query systems: [Technical report]. Department of Informatica E Sistemistica. Universtia Degli Studi Di Roma "La Sapienzaq", 1991
- 3 Balkir N H, Ozsoyoglu G, Ozsoyoglu Z M. A Graphical Query Language: VISUAL. Department of Computer Engineering and Science Case Western Reserve University, 1997
- 4 Kaushik S, Rundensteiner E A. SVIQUQL: A Spatial Visual Query and Exploration Language. Department of Computer Science Worcester Polytechnic Institute, 1997
- 5 Aslandogan Y A. VQL: A Visual Query Language for Uniform Database Access. Department of Computer Engineering and Science Case Western Reserve University, University of Maryland, 1995
- 6 Angelaccio M, Catarci T, Santucci G. QBD: a Graphical Query Language with Recursion. Dipartimento di Informatica e Sistemistica Universita' degli Studi di Roma, 1990
- 7 Fuhr N, Groβjohann K. XIRQL: A Query Language for Information Retrieval in XML Documents. University of Dortmund, Germany, 2001
- 8 W3C. XML-QL: A Query Language for XML. <http://www.w3.org/TR/NOTE-xml-ql>
- 9 Ceri S, Comai S, Damiani E, et al. XML-GL: a Graphical Language for Querying and Restructuring XML Documents, 1999
- 10 Hwang E, Subrahmanian V S. Querying Video Libraries. Department of Computer Science, Institute for Advanced Computer Studies, Institute for Systems Research, 1996
- 11 Li J Z, Ozsu M T, Szafron D, Oria V. MOQL: A Multimedia Object Query Language. Department of Computing Science, University of Allberta, 1997
- 12 Oomoto E, Tanaka K. OVID: Design and implementation of a video-object database system. IEEE Trans. on Knowledge and Data Engineering, 1993, 5(4): 629~643
- 13 Graves A, Lalmas M. Video Retrieval using an MPEG7 Based Inference Network. Department of Computer Science Queen Mary, University of London
- 14 Rui Y, Huang T, Mehrotra S. Exploring Video Structure Beyond the Shots. IEEE Intl. Conf. on Multimedia Computing and Systems, 1998
- 15 Rui Y, Huang T, Mehrotra S. Table of Content Analysis in Videos. ACM Journal of Multimedia Systems, Sept. 1999 (Special Issue on Video Library)
- 16 Working Documents for MPEG-7. http://mpeg.telecomitalia.com/working_documents.htm
- 17 MPEG MDS Group. Text of ISO/IEC 15938-5 FDIS Information Technology -Multimedia Content Description Interface - Part 5 Multimedia Description Schemes. ISO/IEC JTC1/SC29/WG11 MPEG01/N4242, Sydney, July 2001