An MPEG-7 Compatible Video Retrieval System with Integrated Support for Complex Multimodal Queries

Muhammet Baştan, Hayati Çam, Uğur Güdükbay, Özgür Ulusoy
Bilkent University
Department of Computer Engineering
Ankara, Turkey
Tel: +90 312 290 1261, Fax: +90 312 266 4047
(bastan,hayati,gudukbay,oulusoy)@cs.bilkent.edu.tr

September 14, 2009

Abstract

We present *BilVideo-7*, an MPEG-7 compatible, video indexing and retrieval system that supports complex multimodal queries in a unified framework. An MPEG-7 profile is developed to represent the videos by decomposing them into *Shots, Keyframes, Still Regions* and *Moving Regions*. The MPEG-7 compatible XML representations of videos according to this profile are obtained by the MPEG-7 compatible video feature extraction and annotation tool of *BilVideo-7*, and stored in a native XML database. Users can formulate *text-based semantic, color, texture, shape, location, motion* and *spatio-temporal* queries on an intuitive, easy-to-use *Visual Query Interface*, whose *Composite Query Interface* can be used to specify very complex queries containing any type and number of video segments with their descriptors. The multi-threaded *Query Processing Server* parses incoming queries into subqueries and executes each subquery in a separate thread. Then, it fuses subquery results in a bottom-up manner to obtain the final query result. The whole system is unique in that it provides very powerful querying capabilities with a wide range of descriptors and multimodal query processing in an MPEG-7 compatible interoperable environment. We present sample queries to demonstrate the capabilities of the system.

Keywords: MPEG-7, video retrieval, feature extraction, annotation, multimodal query processing.

1 Introduction

Early prototype multimedia database management systems used the query-by-example (QBE) paradigm to respond to user queries. Users needed to formulate their queries by providing examples or sketches. The Query-by-keyword (QBK) paradigm, on the other hand, has emerged due to the desire to search multimedia content in terms of semantic concepts using keywords or sentences rather than low-level multimedia descriptors. This is because it is much easier to formulate some queries by keywords, which is also the way text retrieval systems work. However, some queries are still easier to formulate by examples or sketches (e.g., the trajectory of a moving object). Moreover, there is the so-called "semantic gap" problem, the disparity between low-level representation and high-level semantics, which makes it very difficult to build multimedia systems capable of supporting keyword-based semantic queries effectively with an acceptable number of semantic concepts. The consequence is the need to support both query paradigms in an integrated way so that users can formulate queries containing both high-level semantic and low-level descriptors.

Another important issue to be considered in today's multimedia systems is interoperability. This is especially crucial for distributed architectures if the system is to be used by multiple heterogeneous clients. Therefore, MPEG-7 [1] standard as the multimedia content description interface can be employed to address this issue.

The design of a retrieval system is directly affected by the type of queries to be supported. Types of descriptors and the granularity of the representation determine the system's performance in terms of speed and accuracy. Below, we give some example video query types that might be attractive for most users, but which also are not supported by the existing systems all together in an MPEG-7 compatible framework.

- *Content-based queries by examples.* The user may specify an image, an image region or a video segment and the system returns video segments similar to the input query.
- *Text-based semantic queries.* Queries may be specified by a set of keywords corresponding to high-level semantic concepts and relations between them.
- *Spatio-temporal queries.* Queries related to spatial and temporal locations of objects and video segments within the video.
- *Composite queries.* These queries may contain any combination of other simple queries. The user composes the query (hence the name 'composite' query) by putting together image/video segments and specifying their properties, and then asks the system to retrieve

similar ones from the database. This type of queries is especially desirable to formulate very complex queries easily.

We developed *BilVideo-7* as a comprehensive, MPEG-7 compatible video database system to support such multimodal queries in a unified video indexing and retrieval framework. We designed an MPEG-7 profile for video representation which enables detailed queries on videos, and used our MPEG-7 compatible video feature extraction and annotation tool to obtain the MPEG-7 compatible video representations according to this profile. The *Visual Query Interface* of *BilVideo-7* is an easy-to-use and powerful query interface to formulate complex multimodal queries easily, with support for a comprehensive set of MPEG-7 descriptors. Queries are processed on the multi-threaded *Query Processing Server* with a multimodal query processing and subquery result fusion architecture, which is also suitable for parallelization.

The name *BilVideo-7* is reminiscent of BilVideo [2], which was a prototype video database system that supported keyword-based spatio-temporal queries using a knowledge-base and a Prolog inference engine. *BilVideo-7* is a new system, developed from scratch and is different from BilVideo in terms of MPEG-7 compatibility (BilVideo was not MPEG-7 compatible), video data model, multimodal query processing, query formulation capabilities and wide range of MPEG-7 descriptors supported.

2 Related Work

2.1 MPEG-7 Standard

MPEG-7 [1] is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the standards MPEG-1, MPEG-2 and MPEG-4. Different from the previous MPEG standards, MPEG-7 is designed to describe the content of multimedia. It is formally called "Multimedia Content Description Interface."

MPEG-7 offers a comprehensive set of audiovisual description tools in the form of Descriptors (D) and Description Schemes (DS) that describe the multimedia data, forming a common basis for applications. The Description Definition Language (DDL) is based on W3C XML with some MPEG-7 specific extensions, such as vectors and matrices. Therefore, MPEG-7 documents are XML documents that conform to particular MPEG-7 schemas for describing multimedia content. Descriptors describe features, attributes or groups of attributes of multimedia content. Description Schemes describe entities or relationships pertaining to multimedia content. They specify the structure and semantics of their components, which may be Description Schemes, Descriptors or data types.

The eXperimentation Model (XM) [3] is the software for all the reference code of the MPEG-7 standard. It implements the normative components of MPEG-7. MPEG-7 standardizes multimedia content description but it does not specify how the description is produced. It is up to the developers of MPEG-7 compatible applications how the descriptors are extracted from the multimedia, provided that the output conforms to the standard. MPEG-7 Visual Description Tools consist of basic structures and Descriptors that cover the following basic visual features for multimedia content: *color, texture, shape, motion, localization,* and *face recognition* [1].

- *Color Descriptors:* Color Structure Descriptor (CSD), Scalable Color Descriptor (SCD), Dominant Color Descriptor (DCD), Color Layout Descriptor (CLD), Face Recognition Descriptor (FRD), Group-of-Frame or Group-of-Picture Descriptor (GoF/GoP).
- *Texture Descriptors:* Edge Histogram Descriptor (EHD), Homogeneous Texture Descriptor (HTD), Texture Browsing Descriptor (TBD).
- *Shape Descriptors:* Contour Shape Descriptor (CShD), Region Shape Descriptor (RSD).
- *Motion Descriptors:* Motion Activity (MAc), Motion Trajectory (MTr), Camera Motion, Parametric Motion.
- *Localization Descriptors:* Region Locator, Spatio-temporal Locator.

In MPEG-7, the semantic content of multimedia (e.g., objects, events, concepts) can be described by text annotation (free text, keyword, structured) and/or semantic entity and semantic relation tools. Free text annotations describe the content using unstructured natural language text (e.g., Barack Obama visits Turkey in April). Such annotations are easy for humans to understand but difficult for computers to process. Keyword annotations use a set of keywords (e.g., Barack Obama, visit, Turkey, April) and are easier to process by computers. Structured annotations strike a balance between simplicity (in terms of processing) and expressiveness. They consist of elements each answering one of the following questions: who, what object, what action, where, when, why and how (e.g., who: Barack Obama, what action: visit, where: Turkey, when: April).

More detailed descriptions about semantic entities such as objects, events, concepts, places and times can be stored using semantic entity tools. The semantic relation tools describe the semantic relations between semantic entities using the normative semantic relations standardized by MPEG-7 (e.g., agent, agentOf, patient, patientOf, result, resultOf, similar, opposite, user, userOf, location, locationOf, time, timeOf) or by non-normative relations [1].

The semantic tools of MPEG-7 provide methods to create very brief or very extensive semantic descriptions of multimedia content. Some of the descriptions can be obtained automatically while most of them require manual labeling. Transcribed text obtained from automatic speech recognition (ASR) tools can be used as free text annotations to describe video segments. Keyword and structured annotations can be obtained automatically to some extent using state-of-the-art auto-annotation techniques. Description of semantic entities and relations between them cannot be obtained automatically with the current-state-of-the-art, therefore, considerable amount of manual work is needed for this kind of semantic annotation.

In 2007, MPEG adopted a query format, MPEG Query Format (MPQF) [4], to provide a standard interface between clients and MPEG-7 databases for multimedia content retrieval systems. The query format is based on XML and consists of three main parts: (1) Input query format defines the syntax of query messages sent by a client to the server and supports different types of queries: query by free text, query by description, query by XQuery, spatial query, temporal query, etc. (2) Output query format specifies the structure of the result set to be returned. (3) Query management tools are used to search and choose the desired services for retrieval.

2.2 MPEG-7 Compatible Systems

Although MPEG-7 was published in 2001, only a few MPEG-7 compatible multimedia systems have been developed so far. The comprehensiveness and flexibility of MPEG-7 allow its usage in a broad range of applications, but also increase its complexity and adversely affect interoperability. To overcome this problem, profiling has been proposed. An MPEG-7 profile is a subset of tools defined in MPEG-7, providing a particular set of functionalities for one or more classes of applications. In [5], an MPEG-7 profile is proposed for detailed description of audiovisual content that can be used in a broad range of applications.

An MPEG-7 compatible Database System extension to Oracle DBMS is proposed in *MPEG-7 MMDB* [6]. The resulting system is demonstrated by audio and image retrieval applications. In [7], algorithms for the automatic generation of three MPEG-7 DSs are proposed: (1) *Video Table of Contents DS*, for active video browsing, (2) *Summary DS*, to enable the direct use of meta data annotation of the producer, and (3) *Still Image DS*, to allow interactive content-based image retrieval. Tseng *et al.* [8] address the issues associated with designing a video personalization and summarization system in heterogeneous environments utilizing MPEG-7 and MPEG-21.

IBM's *VideoAnnEx Annotation Tool* [9] enables users to annotate video sequences with MPEG-7 metadata. Each shot is represented by a single keyframe and can be annotated with static scene descriptions, key object descriptions, event descriptions and other custom lexicon sets that may be provided by the user. The tool is limited to concept annotation and cannot extract low-level MPEG-7 descriptors from the video. The *M-OntoMat-Annotizer* [10] software tool aims at linking low-level MPEG-7 visual descriptions to conventional Semantic Web ontologies and annotations. The visual descriptors are expressed in *Resource Description Framework (RDF)*. The IFINDER system [11] is developed to produce limited MPEG-7 representation from audio and video by speech processing, keyframe extraction and face detection. *ERIC7* [12] is a software test-bed that implements Content-Based Image Retrieval (CBIR) using image-based MPEG-7 color, texture and shape descriptors. *Caliph & Emir* [13] are MPEG-7 based Java prototypes for digital photo and image annotation and retrieval, supporting graph-like annotations for semantic meta data and content-based image retrieval using MPEG-7 descriptors. Cao et al. [14] describe a middleware solution to access a bundle of MPEG-7 based multimedia services from mobile devices.

The MPEG-7 compatible systems described above have two major problems. (1) Most of them use a coarse image or video representation, extracting low-level descriptors from whole images or video frames and annotating them, but ignoring region-level descriptors. This coarse representation in turn limits the range of queries. (2) The user cannot perform complex multimodal queries by combining several video segments and descriptors in different modalities. *BilVideo-7* addresses these two major problems by adopting an MPEG-7 profile with a more detailed video representation (Section 3) and using a multimodal query processing and bottom-up subquery result fusion architecture to support complex multimodal queries (e.g., composite queries – see Section 6 for examples) with a comprehensive set of MPEG-7 descriptors.

3 MPEG-7 Compatible Representation of Video

The first step in constructing an MPEG-7 compatible video management system is to decide what kind of queries will be supported and then to design an MPEG-7 profile accordingly. The representation of video is crucial since it directly affects the system's performance. There is a trade-off between the accuracy of representation and the speed of access: more detailed representation will enable more detailed queries but will also result in longer response time during retrieval. Keeping these factors in mind, we adapted our MPEG-7 profile from the one described in [5] to represent image, audio and video collections. Our profile corresponds to the video representation portion of the detailed audiovisual profile, with our own interpretation of what to represent with *Keyframes, Still* and *Moving Regions* so that our system can support the wide range of queries it is designed for. First, audio and visual data are separated (Media Source Decomposition [1]). Then, visual content is hierarchically decomposed into smaller structural and semantic units. An example of video decomposition according to this profile is shown in Figure 1.

Temporal Decomposition of video into shots. Video is partitioned into non-overlapping video segments called shots (sequence of frames captured by a single camera in a single continuous action), each having a temporal location (start time, duration), semantic annotation to describe the objects and/or events with free text, keyword and structured annotations, and visual descriptors (e.g., motion, GoF/GoP).

Temporal Decomposition of shots. The background content of the shots does not change much, especially if the camera is not moving. This static content can be represented by a single keyframe or a few keyframes. Therefore, each shot is decomposed into smaller, more homogeneous video segments (keysegments) which are represented by keyframes. Each keyframe is described by a temporal location, semantic annotations and a set of visual descriptors. The visual descriptors are extracted from the frame as a whole.

Each keyframe is also decomposed into a set of *Still Regions* (*Spatio-temporal Decomposition*) to keep more detailed region-based information in the form of spatial location by the MBRs of the region, semantic annotation and region-based visual descriptors.

Spatio-temporal Decomposition of shots into Moving Regions. Each shot is also decomposed into a set of *Moving Regions* to represent the dynamic and more important content of the shots corresponding to the salient objects. Hence, more information can be stored for *Moving Regions* to enable more detailed queries about salient objects. We represent all salient objects with *Moving Regions* even if they are not moving. *Faces* are also represented by *Moving Regions*, having an additional visual descriptor: Face Recognition Descriptor.

Since the position, shape, motion and visual appearance of the salient objects may change throughout the shot, descriptors sampled at appropriate time points should be stored. The trajectory of an object is represented by the *Motion Trajectory* descriptor. The MBRs and visual descriptors of the object throughout the shot are stored by temporally decomposing the object into *Still Regions.*

Video Segment: From here on, we refer to *Shots, Keyframes, Still Regions* and *Moving Regions*, as *video segments.*

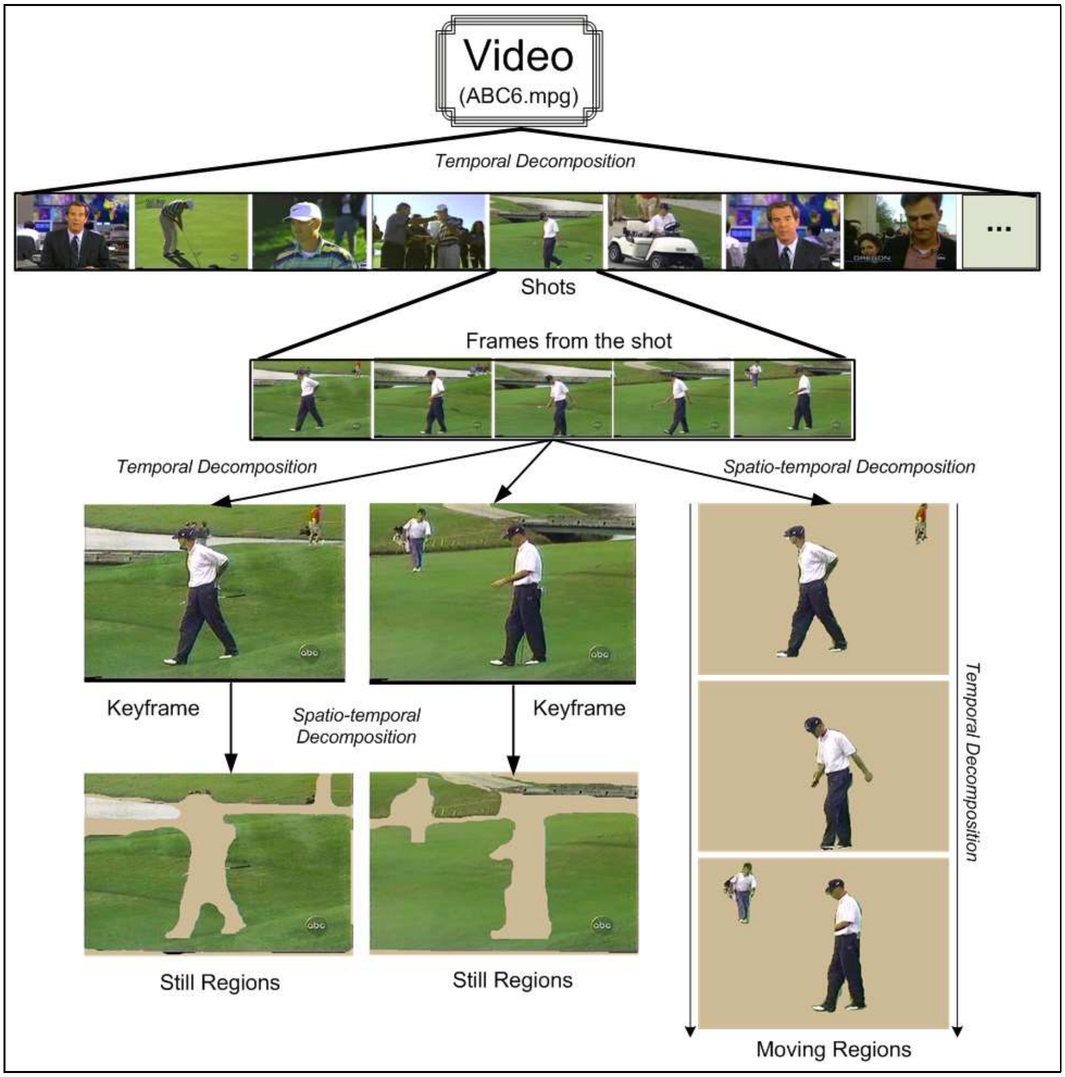


Figure 1: MPEG-7 decomposition of a video according to the MPEG-7 profile used in *BilVideo-7*. Low-level color, texture and shape descriptors of the *Still* and *Moving Regions* are extracted from the selected arbitrarily shaped regions, but the locations of the regions are represented by their Minimum Bounding Rectangles (MBR).

4 System Architecture

BilVideo-7 has a client-server architecture (Figure 2). Users formulate queries on *BilVideo-7* clients, which communicate with the *BilVideo-7 Query Processing Server* using an XML-based query language [15] over TCP/IP. The *Query Processing Server* parses queries into subqueries, retrieves the required data from the XML database using XQuery for each subquery, executes subqueries, fuses the results of subqueries and sends the results back to the clients.

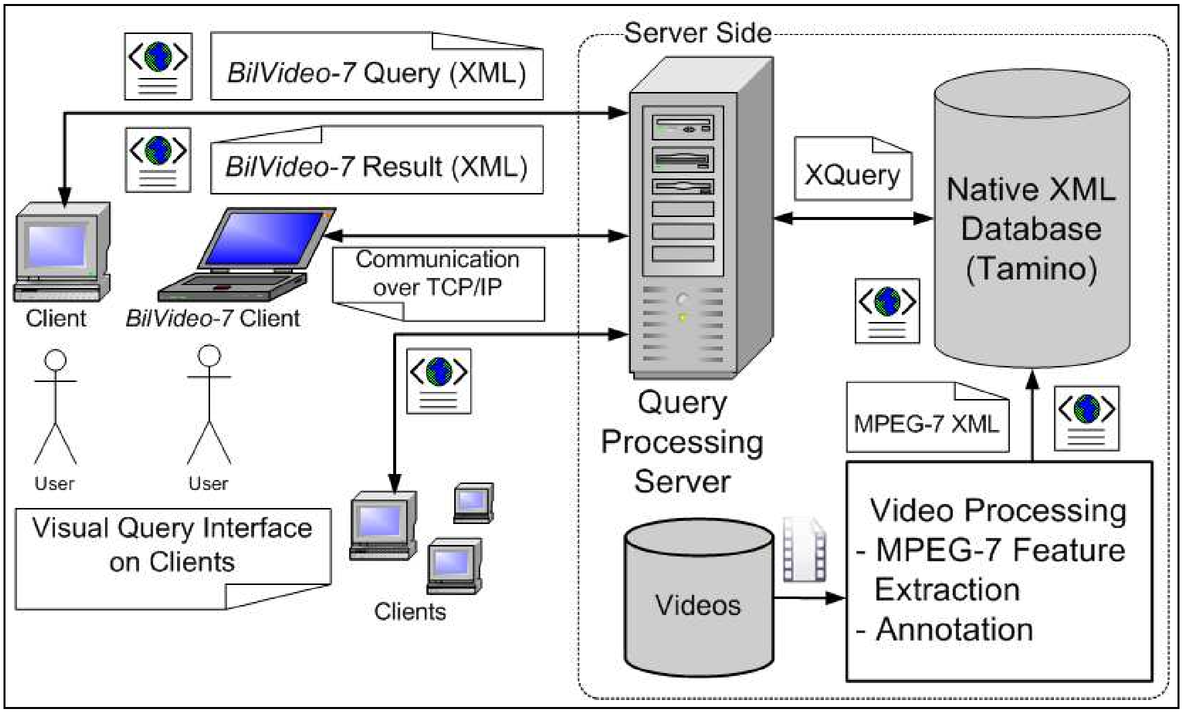
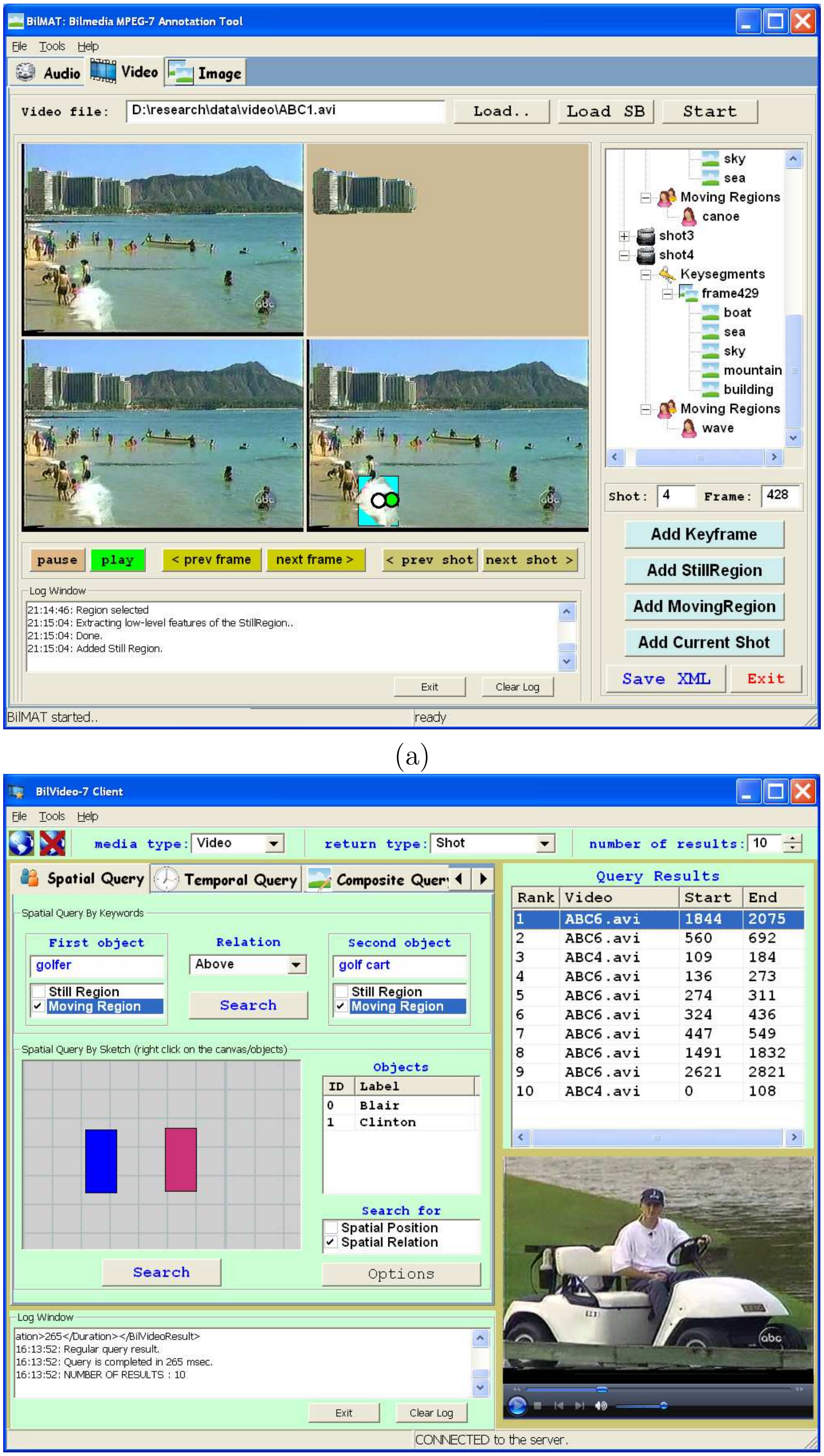


Figure 2: Client-server architecture of *BilVideo-7*.

4.1 MPEG-7 Compatible Feature Extraction and Annotation

MPEG-7 representations of videos are obtained using the MPEG-7 compatible video feature extraction and annotation tool (cf. Figure 3 (a)). In the figure, the current video frame is shown at the top left, latest processed frame is at the bottom left, latest selected region is at the top right, and selected Moving Regions along with their trajectories are at the bottom right. Selected video segments are shown on the right in a hierarchical tree view reflecting the structure of the video.

Videos, along with shot boundary information, are loaded and then processed on a shot-by-shot basis. Users can manually select *Keyframes*, *Still Regions* and *Moving Regions* and then annotate the *Video, Shots, Keyframes, Still Regions* and *Moving Regions* with free text, keyword and structured annotations. The MPEG-7 visual descriptors (color, texture, shape, motion, localization) for the selected video segments are computed by the tool, using an MPEG-7 feature extraction library adapted from MPEG-7 XM Reference Software [3]. The semantic content is described by text annotations (free text, keyword and structured annotation), which strike a good balance between simplicity (in terms of manual annotation effort and processing during querying) and expressiveness.



Video file: D:\research\data\video\ABC1.avi

Shot: 4 Frame: 428

```
21:14:46: Region selected
21:15:04: Extracting low-level features of the StillRegion..
21:15:04: Done.
21:15:04: Added Still Region.
```

BilMAT started.. ready

(a)

media type: Video return type: Shot number of results: 10

First object: golfer
☐ Still Region
☑ Moving Region

Relation: Above

Second object: golf cart
☐ Still Region
☑ Moving Region

Objects

ID	Label
0	Blair
1	Clinton

Search for
☐ Spatial Position
☑ Spatial Relation

Query Results

Rank	Video	Start	End
1	ABC6.avi	1844	2075
2	ABC6.avi	560	692
3	ABC4.avi	109	184
4	ABC6.avi	136	273
5	ABC6.avi	274	311
6	ABC6.avi	324	436
7	ABC6.avi	447	549
8	ABC6.avi	1491	1832
9	ABC6.avi	2621	2821
10	ABC4.avi	0	108

```
ation>265</Duration></BilVideoResult>
16:13:52: Regular query result.
16:13:52: Query is completed in 265 msec.
16:13:52: NUMBER OF RESULTS : 10
```

CONNECTED to the server.

(b)

Figure 3: (a) MPEG-7 compatible video feature extraction and annotation tool. (b) *BilVideo-7* client *Visual Query Interface*. This screenshot shows the *Spatial Query Interface*. For the other query interfaces, please see [15].

The output is saved as an MPEG-7 compatible XML file for each video. We use a native XML database, Tamino [16], to store the MPEG-7 representations. Native XML databases use the XML data model directly, and thus, it is more convenient and natural to use a native XML database, since no mapping and conversion to other data models is required and it is very easy to set up the database using the publicly available MPEG-7 schema.

4.2 Visual Query Interface

Users formulate queries on *BilVideo-7* clients' *Visual Query Interface*, which provides an intuitive, easy-to-use query formulation interface and consists of several tabs, each for a different type of query with a comprehensive set of descriptors and query options. As shown in Figure 3 (b), the query formulation tabs are on the left, the query result list is displayed at the top right, the query results can be viewed on the media player at the bottom right, and messages are displayed at the bottom left. The user can select the media type, return type and maximum number of results to be returned, from the toolbar at the top. The queries are converted into *BilVideo-7Query* format [15] in XML and sent to the *BilVideo-7 Query Processing Server*.

Video Table of Contents (VideoToC) is a useful facility to let the user browse through the video collection in the database, to see the contents of each video in a hierarchical tree view reflecting the structure of the MPEG-7 representation of the video in XML format and to see the high-level semantic concepts in the collection and in each video separately. The user can browse through each video in the collection and see all the *Shots, Keyframes, Still Regions* and *Moving Regions* as well as the semantic concepts they are annotated with and their temporal location (Media Time) in the video.

Textual Query Interface enables the user to formulate high-level semantic queries quickly by entering keywords and specifying the type of video segment (*Shot, Keyframe, Still Region, Moving Region*) and annotation (free text, keyword, structured) to search in.

Color, Texture, Shape Query Interface is used for querying video segments by MPEG-7 color, texture and shape descriptors. The input media can be a video segment, a whole image or an image region. The descriptors need to be extracted from the selected input media. Instead of uploading the input media to the server and extracting the descriptors there, we extract the descriptors on the client, form the XML-based query expression containing the descriptors and send the query to the server. Therefore, the MPEG-7 feature extraction module is integrated with *BilVideo-7* clients. The user also specifies the type of video segments to search in, and also other query options, such as weights and thresholds for each type of descriptor. See sections 5.1 and 5.2 for the details of query processing.

Motion Query Interface is for the formulation of Motion Activity and Motion Trajectory queries. Trajectory points are entered using the mouse. The user can optionally specify keywords for the *Moving Region* for which the trajectory query will be performed.

Spatial Query Interface enables the user to formulate spatial queries for *Still* and *Moving Regions* using either keywords and a set of predefined spatial relations (left, right, above, below, east, west, etc.) or by sketching the minimum bounding rectangles (MBR) of objects using the mouse, and if desired, giving labels to them. It is possible to query objects based on location, spatial relations or both.

Temporal Query Interface is very similar to spatial query interface; this time, the user specifies temporal relations between video segments (*Shots, Keyframes, Still Regions, Moving Regions*) either by selecting from a predefined list (before, after, during, etc.) or by sketching the temporal positions of the segments using the mouse.

Composite Query Interface is the most powerful query interface and enables the user to formulate very complex queries easily. The query is composed by putting together any number of *Shots, Keyframes, Still Regions* and *Moving Regions* and specifying their properties as text-based semantic annotations, visual descriptors, location, spatial and temporal relations. Using this interface, the user can describe a video segment or a scene and ask the system to retrieve similar video segments.

XQuery Interface is more suited to experienced users who can write XQueries to search in the database. This is the most flexible interface and the user can specify a wide range of queries.

5 Query Processing

Query processing is done on the *Query Processing Server*, which is a multi-threaded server side component that listens to a configured TCP port and accepts incoming clients and processes their queries. Clients send queries in the XML-based *BilVideo-7Query* format [15] and receive query results in XML-based *BilVideo-7Result* format, which contains a list of video segments (video name, start time, end time) in ranked order. Current version of BilVideo-7 does not support MPQF query language since it is not possible to formulate some of the BilVideo-7 queries in MPQF (e.g., spatial queries by location).

5.1 Multi-threaded Query Execution

Each incoming query is parsed into subqueries and executed in a multi-threaded fashion, with one thread for each type of subquery, as shown in Figure 4 (a). Queries with the same subquery type are accumulated in a queue and executed on a first-in-first-out (FIFO) basis. For example, subqueries for color descriptors (CSD, SCD, DCD, etc.) are added to the end of queue of *Color Query Executor* thread and executed in this order. One XQuery is formed and executed for each subquery, consisting of a single video segment and a single descriptor (e.g., *Keyframe* with CSD). The XML database returns the XQuery results in XML format, which are parsed to extract the actual data (the descriptors).

Textual queries are the easiest to execute since the XML database can handle textual queries and no further processing is needed for the similarity computation. However, the database cannot handle the similarity queries for low-level descriptors. That is, the similarity between a query descriptor and a descriptor in the database cannot be computed by the database. Therefore, the corresponding query execution thread retrieves the relevant descriptors from the database for the video segment in the subquery (e.g., CSD for *Keyframes*) and computes their distances to the query.

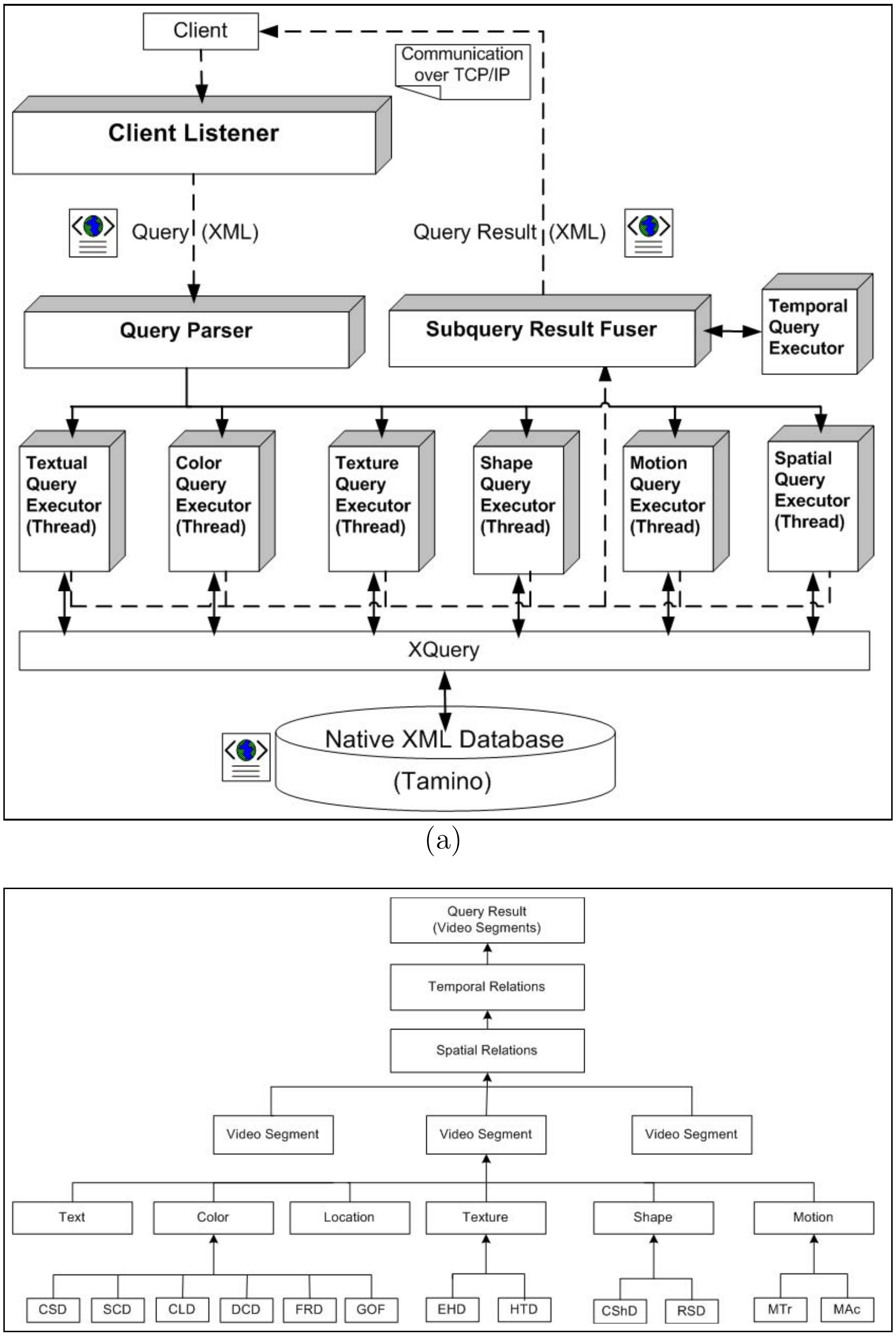
Distance Computations. The distance measures suggested by MPEG-7 are implemented in MPEG-7 XM Reference Software [3] but they are not normative. The evaluation of the distance measures for a set of MPEG-7 descriptors, presented in [17], shows that although there are better distance measures (e.g., pattern difference, Meehl index), the MPEG-7 recommendations are among the best. Therefore, we adapted the distance measures from the XM software. Below, we briefly describe the distance metrics adapted [1, 3]. Q refers to a descriptor in the query, D to a descriptor in the database and d is the computed distance between the descriptors.

L_1-norm is used to compute the distance between Color Structure, Scalable Color, GoF/GoP and Region Shape descriptors.

$$d_{L_1}(Q, D) = \sum_i |Q(i) - D(i)|$$

The distance between two Color Layout descriptors, $Q = \{QY, QCr, QCb\}$ and $D = \{DY, DCr, DCb\}$, is computed by

$$d(Q, D) = \sqrt{\sum_i w_{yi}(QY_i - DY_i)^2} + \sqrt{\sum_i w_{bi}(QCb_i - DCb_i)^2} + \sqrt{\sum_i w_{ri}(QCr_i - DCr_i)^2}$$


Client
Communication over TCP/IP
Client Listener
Query (XML)
Query Result (XML)
Query Parser
Subquery Result Fuser
Temporal Query Executor
Textual Query Executor (Thread)
Color Query Executor (Thread)
Texture Query Executor (Thread)
Shape Query Executor (Thread)
Motion Query Executor (Thread)
Spatial Query Executor (Thread)
XQuery
Native XML Database (Tamino)
Query Result (Video Segments)
Temporal Relations
Spatial Relations
Video Segment
Video Segment
Video Segment
Text
Color
Location
Texture
Shape
Motion
CSD
SCD
CLD
DCD
FRD
GOF
EHD
HTD
CShD
RSD
MTr
MAc

(a)

(b)

Figure 4: (a) The framework of the *Query Processing Server*. Each type of subquery is executed in a separate thread. (b) Subquery results are fused in a bottom-up manner.

where i represents the zigzag-scanning order of the coefficients and the weights (w_{yi}, w_{bi}, w_{ri}) are used to give more importance to the lower frequency components of the descriptor.

The distance between the Edge Histogram descriptors Q and D is computed by adapting the

L_1-norm as

$$d(Q,D) = \sum_{i=0}^{79} |h_Q(i) - h_D(i)| + 5\sum_{i=0}^{4} \left|h_Q^g(i) - h_D^g(i)\right| + \sum_{i=0}^{64} \left|h_Q^s(i) - h_D^s(i)\right|$$

where $h_Q(i)$ and $h_D(i)$ represent the histogram bin values of the descriptors Q and D, $h_Q^g(i)$ and $h_D^g(i)$ for global edge histograms, and $h_Q^s(i)$ and $h_D^s(i)$ for semi-global edge histograms.

The distance between two Homogeneous Texture descriptors Q and D (full layer – using both energy and energy deviation) is computed by

$$\begin{aligned} d(Q,D) = w_{dc}|Q(0) - D(0)| + w_{std}\,|Q(1) - D(1)| + \\ \sum_{n=0}^{RD-1} \sum_{m=0}^{AD-1} w_e(n)\,|Q(n \cdot AD + m + 2) - D(n \cdot AD + m + 2)| + \\ w_{ed}(n)\,|Q(n \cdot AD + m + 32) - D(n \cdot AD + m + 32)| \end{aligned}$$

where w_{dc}, w_{std}, w_e and w_{ed} are weights; the *Radial Division*, $RD = 5$ and *Angular Division*, $AD = 6$. Matching with this distance metric is not scale and rotation invariant.

The distance between two face recognition descriptors Q and D is computed as follows.

$$d(Q,D) = \sum_{i=0}^{47} w_i(Q(i) - D(i))^2$$

where w_i is the weight for the i^{th} descriptor value.

The intensity of Motion Activity is a scalar value, therefore, the distance is computed simply by taking the difference between two descriptor values. When the spatial localization of motion activity is given, Euclidean distance is used. For spatial position queries, Euclidean distance between the center points of objects' MBRs is used. Due to space considerations, we omit the distance metrics for Dominant Color, Contour Shape and Motion Trajectory.

If multiple instances of a descriptor are available for a *Moving Region* to account for the change in its appearance throughout the shot, the distance is computed for all the instances and the minimum is taken. If the computed distance for a video segment in the database is greater than the user-specified distance threshold for the query video segment and descriptor (e.g., for *Keyframe* with CSD, if $d(Q,D)/d_{max} > T_{Keyframe,CSD}$), that segment is discarded. Otherwise, the similarity, $s(Q, D)$, between two descriptors Q and D is computed as

$$s(Q, D) = 1 - d(Q, D)/d_{max}, \quad 0 \leq s(Q, D) \leq 1.0$$

where $d(Q, D)$ is the distance between descriptors Q and D, d_{max} is the maximum possible distance for the type of descriptor in the computation. The maximum distance for each descriptor is computed by taking the maximum distance from a large set of descriptors extracted from video segments.

Spatial Query Processing. Spatial locations of Still Regions and Moving Regions are stored in the database by their MBRs, without any preprocessing to extract and store the spatial relations between them. Therefore, spatial similarity between regions is computed at query execution time. This is computationally more expensive but it provides a more flexible and accurate matching for spatial position and spatial relation queries.

For each *Still Region* or *Moving Region* in the query, first, queries related to the properties of the region (textual, color, texture, shape, location, motion) are executed as described above. Then, the resulting video segments undergo spatial query processing to compute the spatial similarities between them. We use the spatial similarity matching approach described in [18] because of its efficiency and robustness. First, the vectors connecting the center points of objects' MBRs, $\overrightarrow{Q_{xy}}$ and $\overrightarrow{D_{ij}}$, are computed as shown in Figure 5. Then, the pairwise spatial similarity is computed by the cosine of the angle between the vectors $\overrightarrow{Q_{xy}}$ and $\overrightarrow{D_{ij}}$, using vector dot product:

$$d(Q_{xy}, D_{ij}) = \cos\theta = \frac{\overrightarrow{Q_{xy}} \bullet \overrightarrow{D_{ij}}}{\left|\overrightarrow{Q_{xy}}\right| \left|\overrightarrow{D_{ij}}\right|}, \quad 0 \leq \theta \leq \pi, \quad -1 \leq d(Q_{xy}, D_{ij}) \leq +1$$

The output value is in the range [-1, +1], with +1 indicating identical spatial relation and -1 opposite spatial relation. In Figure 5, the spatial relation between the database objects D_1 and D_3 is the most similar to the spatial relation between query objects Q_1 and Q_2.

The text-based spatial queries (*right, left, above, below, etc.*) are executed in the same way, by converting each spatial relation query to a unit vector (Figure 5, left). For instance, Q_x *right* Q_y (Q_x is to the right of Q_y) query is converted to a query vector $\overrightarrow{Q_{xy}}= [-1, 0]$, from Q_x to Q_y.

Multiple MBRs are stored in the database for *Moving Regions* to keep track of their locations. The spatial similarity is computed for all the MBRs and the maximum similarity value is taken as the final similarity.

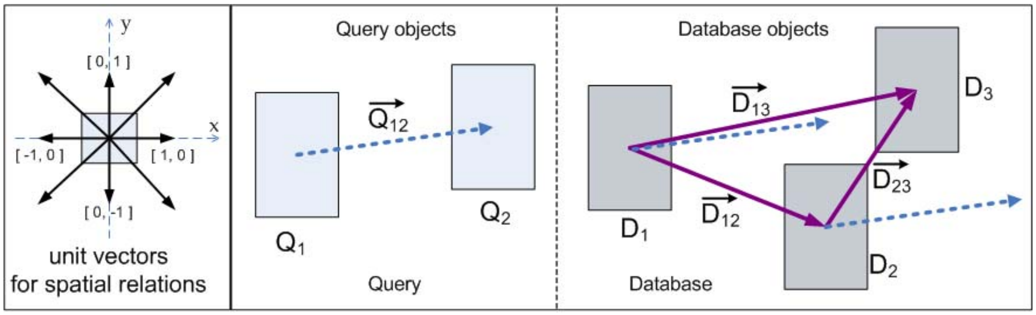


Figure 5: Spatial query processing by vector dot product between the vectors connecting centers of objects' MBRs.

Temporal queries, if any, are executed after spatial queries by checking if the list of video segments satisfies the temporal relations specified in the query. Spatial queries implicitly enforce a temporal relation between *Still* and *Moving Regions*, since they must co-appear on a scene for a certain time interval in the video to satisfy the spatial relations.

5.2 Fusion of Subquery Results

When multiple descriptors, possibly in different modalities, are specified for a query video segment, each is executed as a separate subquery, resulting in a list of video segments with similarity values. These subquery results must be fused to come up with the final query result. This is done in a bottom-up manner as shown in Figure 4 (b). Each node in the tree has an associated weight and threshold, which can be specified by the user during query specification. The similarity at each node is computed as the weighted average of the similarities of its children and the fusion process continues upward in the tree until the final query result is obtained.

To illustrate the fusion process, consider a composite query consisting of a *Keyframe* with color (CSD and DCD), texture (EHD and HTD) and text-based semantic (keyword *golf green*) descriptors. The query processor parses this query into 5 subqueries (CSD, DCD, EHD, HTD and semantic), executes each and produces 5 lists of *Keyframes* from database with similarity values. Then, it fuses color (CSD, DCD) and texture (EHD, HTD) subquery results to obtain the color and texture similarities of each *Keyframe*.

$$
\begin{aligned}
s_{i,Color} &= \frac{w_{Keyframe,CSD}\ s_{i,CSD} + w_{Keyframe,SCD}\ s_{i,SCD}}{w_{Keyframe,CSD} + w_{Keyframe,SCD}} \\
s_{i,Texture} &= \frac{w_{Keyframe,EHD}\ s_{i,EHD} + w_{Keyframe,HTD}\ s_{i,HTD}}{w_{Keyframe,HTD} + w_{Keyframe,HTD}}
\end{aligned}
$$

where $s_{i,Color}$ is the color similarity for the i^{th} *Keyframe*, $w_{Keyframe,CSD}$ is the weight for CSD and so on. If the similarity of Keyframe i is less than the threshold specified by the user, it is discarded. At this point we have 3 lists of *Keyframes* having similarity values for color, texture, semantic (text). We fuse these 3 lists to obtain the final list of *Keyframes*.

$$s_i = \frac{w_{Keyframe,Color}\ s_{i,Color} + w_{Keyframe,Texture}\ s_{i,Texture} + w_{Keyframe,Text}\ s_{i,Text}}{w_{Keyframe,Color} + w_{Keyframe,Texture} + w_{Keyframe,Text}}$$

If there are also spatial or temporal relation subqueries, they are executed and similarity values of the video segments are updated in the same way. Finally, we obtain N_{vs} lists of video segments, where N_{vs} is the number of video segments in the query. The final query result is obtained by fusing these lists using the same weighted average approach as above and sorting the list in descending order of similarity.

6 Experimental Results

The system is implemented in C++. Open Source Computer Vision Library (OpenCV) [19] is used to handle (read, decode, copy, save, etc.) the image and video data. The MPEG-7 compatible video feature extraction and annotation tool uses the MPEG-7 feature extraction library that we adapted from the MPEG-7 XM Reference Software [3].

We performed queries on a video data set consisting of 14 video sequences with $25,000$ frames from TRECVID 2004 and 2008 data sets [20], including news, documentary, educational and archiving program videos. The query result is a list of shots in ranked order, each shown with a representative keyframe in the following figures. For more query examples, please see *BilVideo-7* website [15].

Two spatial queries are shown in Figure 6 (a). The first query at the top searches for the video segments in which a golfer is *above* a golf cart, formulated as a text-based spatial relation query, “golfer *above* golf cart”. The system successfully returns three relevant video segments that exactly match the spatial query condition. The fourth result contains a “golfer” but no “golf cart” and spatial condition is not satisfied. Therefore, its rank is lower than the first three. The second query at the bottom, “Clinton *left* Blair”, is sketch-based. The spatial query condition is satisfied exactly in the first two video segments returned, while it is not satisfied in the last two, but “Clinton” and “Blair” appear together. This is a desirable result of our bottom-up fusion algorithm; as the number of satisfied query conditions for a video segment decreases the video segment’s similarity also decreases, ranking lower in the query result. As a result, the ranking approach is effective and it produces query results that are close to our perception.

Two low-level queries are shown in Figure 6 (b). In the image-based query (top), query image is represented by Color Structure and Dominant Color descriptors and searched in *Keyframes*. In the region-based query (bottom), query region is represented by Color Structure and Region Shape descriptors, and searched in *Moving Regions*. Both query results are satisfactory considering the types of descriptors used.

The three queries shown in Figure 6 (c) are composite queries, in which high-level semantics in the form of keyword annotations and low-level descriptors (DCD, CSD, EHD, RSD) are used together to describe the query video segments. In the first composite query, *Keyframe* is represented by Dominant Color and *golf green*. *Moving Region* is represented by Color Structure, Region Shape and *golfer*. In the second query, two *Still Regions* at the top and at the bottom are represented by Color Structure and Edge Histogram descriptors. The *Moving Region* in the middle is represented by semantic concepts *airplane or boat or helicopter*. Using such queries, the user can access video segments having any specific composition described in the query. The number and type of video segments in the query, as well as the descriptors used to describe them are not limited. This makes the composite queries very flexible and powerful, enabling the user to formulate very complex queries easily. To our knowledge, our system is unique in supporting such complex queries.

Table 1 presents query execution times for several queries. The query execution time is proportional to the number of subqueries (number of video segments and descriptors in the query), database size (number of video segments in the database), the sizes of the descriptors and the complexity of the matching algorithm (distance metric). Queries involving low-level descriptors take longer to execute compared to text-based queries since the distance computations between the low-level descriptors are computationally more expensive. The multi-threaded query processing architecture provides some degree of parallelism and shortens the query execution times when the subqueries are executed in separate threads.

7 Conclusions and Future Work

We described our prototype MPEG-7 compatible video indexing and retrieval system, *BilVideo-7*, that supports different types of multimodal queries in an integrated way. To our knowledge, *BilVideo-7* is the most comprehensive MPEG-7 compatible video database system currently available, in terms of the wide range of MPEG-7 descriptors and manifold query options supported. The MPEG-7 profile used for the detailed representation of the videos enables the sys-

Table 1: Query execution times (in seconds). *Query Processing Server* and Tamino XML database are installed on a notebook PC with Intel Core 2 dual-core 2.0 GHz processors and 2.0 GB of RAM, running Windows XP.

Query type	*Description* *(Segments and descriptors)*	*Execution time (sec)*
Text-based semantic query	Keyframe (keyword)	0.125
Text-based semantic query	Moving Region (keyword)	0.125
Text-based semantic query	Keyframe (keyword), Moving Region (keyword)	0.188
Color query	Keyframe (CSD)	0.141
Texture query	Keyframe (HTD)	0.125
Color + Texture query	Keyframe (CSD+HTD)	0.172
Shape query	Moving Region (RSD)	0.156
Spatial query	Text-based, 2 Still Regions	0.172
Spatial query	Text-based, 2 Moving Regions	0.187
Spatial query	Sketch-based, 2 Moving Regions	0.187
Composite query in Figure 6 (c), top	Keyframe (DCD+keyword), Moving Region (CSD+RSD+keyword)	0.438
Composite query in Figure 6 (c), bottom	2 Still Regions (CSD+EHD), Moving Region (keyword)	0.391

tem to respond to complex queries with the help of the flexible query processing and bottom-up subquery result fusion architecture. The user can formulate very complex queries easily using the *Visual Query Interface*, whose *Composite Query Interface* is novel in formulating a query by describing a video segment as a composition of several video segments along with their descriptors. The broad functionality of the system is demonstrated by sample queries which are handled effectively by the system. The retrieval performance depends very much on the MPEG-7 descriptors and the distance measures used. As a future work, we will investigate distance measures other than the ones recommended by MPEG-7 [17].

The multi-threaded query execution architecture is suitable for parallelization. This is required for video databases of realistic size to keep the response time of the system at interactive rates. In a parallel architecture, each query processing node may keep the data for a subset of descriptions (e.g., text, color, texture, shape) and execute only the relevant subqueries. A central *Query Processor* can coordinate the operation of query processing nodes.

The major bottleneck for the system is the generation of the MPEG-7 representations of videos by manual processing, which is time consuming, error-prone and which also suffers from human subjectivity. This hinders the construction of a video database of realistic size. Therefore, our current focus is on equipping the MPEG-7 compatible video feature extraction and annotation tool with as much automatic processing capabilities as possible to reduce manual processing time, errors and human subjectivity during region selection and annotation.

Finally, future versions of *BilVideo-7* will also support representation and querying of audio and image data. The multimodal query processing architecture makes it easy to add new descriptors for new modalities (e.g., audio descriptors). Images can be considered to be a special case of *Keyframes* which are decomposed into *Still Regions*, and hence can be supported easily.

8 Acknowledgments

The authors would like to thank Rana Nelson for proofreading this manuscript, and anonymous reviewers for their constructive comments.

References

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

[2] M. E. Dönderler, E. Saykol, U. Arslan, Ö. Ulusoy, and U. Güdükbay, "BilVideo: Design and Implementation of a Video Database Management System," *Multimedia Tools and Applications*, vol. 27, no. 1, pp. 79–104, 2005.

[3] "ISO/IEC 15938-6:2003: Information Technology – Multimedia Content Description Interface – Part 6: Reference software," 2009. [Online].
Available: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

[4] "ISO/IEC FCD 15938-12:2007: Information technology – Multimedia content description interface – Part 12: Query Format," 2007. [Online].
Available: http://www.chiariglione.org/mpeg/working_documents/mpeg-07/mp7qf/mp7qf-fcd.zip

[5] W. Bailer and P. Schallauer, "Detailed Audiovisual Profile: Enabling Interoperability Between MPEG-7 Based Systems," *Proceedings of the 12th International Multi-Media Modelling Conference Proceedings*, pp. 217–224, January 2006.

[6] M. Döller and H. Kosch, "The MPEG-7 Multimedia Database System (MPEG-7 MMDB)," *The Journal of Systems and Software*, vol. 81, no. 9, pp. 1559–1580, 2008.

[7] Y. Rui, "MPEG-7 Enhanced Ubi-Multimedia Access – Convergence of User Experience and Technology," in *Proceedings of the First IEEE International Conference on Ubi-Media Computing*, 31 August 2008, pp. 177–183.

[8] B. Tseng, C.-Y. Lin, and J. Smith, "Using MPEG-7 and MPEG-21 for Personalizing Video," *IEEE Multimedia*, vol. 11, no. 1, pp. 42–52, January-March 2004.

[9] "IBM VideoAnnEx Annotation Tool," 2009. [Online].
Available: http://www.research.ibm.com/VideoAnnEx

[10] K. Petridis, D. Anastasopoulos, C. Saathoff, N. Timmermann, Y. Kompatsiaris, and S. Staab, "M-OntoMat-Annotizer: Image Annotation Linking Ontologies and Multimedia Low-Level Features," in *Lecture Notes in Computer Science*, vol. 4253. Springer, 2006, pp. 633–640.

[11] J. Löffler, K. Biatov, C. Eckes, and J. Köhler, "IFINDER: An MPEG-7-Based Retrieval System for Distributed Multimedia Content," in *Proceedings of the Tenth ACM International Conference on Multimedia*, 2002, pp. 431–435.

[12] L. Gagnon, S. Foucher, and V. Gouaillier, "ERIC7: An Experimental Tool for Content-Based Image Encoding and Retrieval Under the MPEG-7 Standard," in *Proceedings of the Winter International Symposium on Information and Communication Technologies*, 2004, pp. 1–6.

[13] M. Lux, J. Becker, and H. Krottmaier, "Caliph&Emir: Semantic Annotation and Retrieval in Personal Digital Photo Libraries," in *Proceedings of the CAiSE '03 Forum at 15th Conference on Advanced Information Systems Engineering*, Velden, Austria, June 2003, pp. 85–89.

[14] Y. Cao, M. Jarke, R. Klamma, O. Mendoza, S.N. Srirama, "Mobile Access to MPEG-7 Based Multimedia Services," in *Proc. of the Tenth International Conference on Mobile Data Management*, Taipei, Taiwan, May 2009, pp. 102-111.

[15] "BilVideo-7: MPEG-7 Compatible Video Database System Web Site," 2009. [Online].
Available: http://www.cs.bilkent.edu.tr/~bilmdg/bilvideo-7

[16] "Tamino: Software AG XML Data Management Product," 2009. [Online].
Available: http://www.softwareag.com/Corporate/products/wm/tamino/default.asp

[17] H. Eidenberger, "Distance measures for MPEG-7-based retrieval," in *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2003, pp. 130–137.

[18] J. Z. Li and M. T. Ozsu, "STARS: A Spatial Attributes Retrieval System for Images and Videos," in *Proceedings of 4th International Conference on Multimedia Modeling*, 1997, pp. 69–84.

[19] "OpenCV: Open Source Computer Vision Library," 2009. [Online].
Available: http://opencvlibrary.sourceforge.net

[20] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 2006, pp. 321–330.

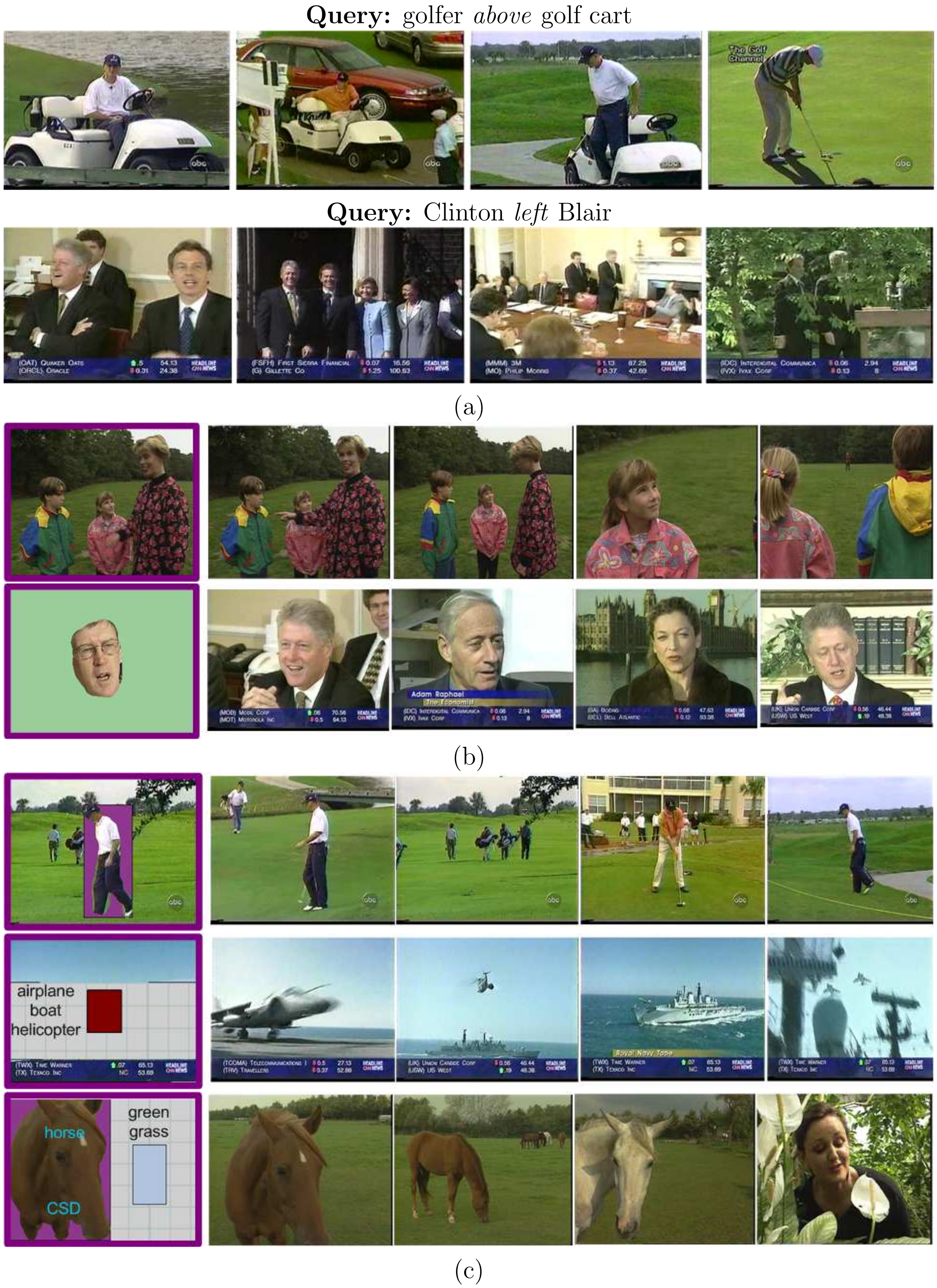


Figure 6: (a) Spatial queries. Top: text-based spatial relation query, "golfer *above* golf cart". Bottom: sketch-based spatial relation query, "Clinton *left* Blair", formulated by drawing two rectangles and labeling them as *Clinton* and *Blair*. (b) Image-based (top) and region-based (bottom) low-level queries (queries are on the left). (c) Composite queries.

9 Author Biographies

Muhammet Baştan is a Ph.D. candidate in the Department of Computer Engineering at Bilkent University, Ankara, Turkey. His research interests include computer vision, pattern recognition, multimedia retrieval, MPEG-7, image/video processing, saliency, segmentation and annotation. He received the M.S. and B.S. degrees from Sabancı University, Istanbul, Turkey and Middle East Technical University, Ankara, Turkey, respectively. Contact him at bastan@cs.bilkent.edu.tr.

Hayati Çam has an M.S. in Computer Engineering from Bilkent University, Ankara, Turkey. His research interests include multimedia databases and computer vision. He received his M.S. in Computer Engineering from Bilkent University in 2008.

Uğur Güdükbay is an associate professor in the Department of Computer Engineering, Bilkent University, Ankara, Turkey. His research interests include multimedia databases, computer vision and computer graphics. He received his Ph.D. in Computer Engineering and Information Science from Bilkent University in 1994. He is a senior member of IEEE and ACM. Contact him at gudukbay@cs.bilkent.edu.tr.

Özgür Ulusoy is a full professor in the Department of Computer Engineering, Bilkent University, Ankara, Turkey. His research interests include multimedia databases, web databases, peer-to-peer systems, data management for mobile systems, and real-time and active database systems. He received his Ph.D. in Computer Science from University of Illinois at Urbana-Champaign in 1992. He is a member of IEEE and ACM. Contact him at oulusoy@cs.bilkent.edu.tr.