BilVideo-7: An MPEG-7-Compatible Video Indexing and Retrieval System

Muhammet Baştan, Hayati Çam, Uğur Güdükbay, and Özgür Ulusoy
Bilkent University

BilVideo-7 is an MPEG-7-compatible, distributed, video indexing and retrieval system that supports complex multimodal queries in a unified framework.

Early prototype multimedia database management systems used the query-by-example paradigm to respond to user queries. Users needed to formulate their queries by providing examples or sketches. The query-by-keyword 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. After all, it's much easier to formulate some queries by using keywords. However, some queries are still easier to formulate by examples or sketches—for example, 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. This gap makes it difficult to build multimedia systems that are capable of effectively supporting keyword-based semantic queries while also being able to interpret an acceptable number of semantic concepts. The consequence is the need to support both query paradigms in an integrated way to provide users with queries containing both high-level semantic concepts and low-level descriptors. Another important issue to be considered is interoperability, which is especially crucial for distributed architectures if the system is to be used by multiple heterogeneous clients. In the system described here, we use the MPEG-7 standard as a multimedia content-description interface to address this issue.[1]

The design of a retrieval system is directly affected by the type of queries to be supported, especially because the types of descriptors and the granularity of the representation determine the system's speed and accuracy. Some example video query types that might be attractive for most users, but which also are not supported by the existing systems in an MPEG-7-compatible framework, include the following:

- *Content-based queries by examples*. The user specifies an image, an image region, or a video segment and the system returns video segments that are similar to the input query.
- *Text-based semantic queries*. Queries are specified by a set of keywords corresponding to high-level semantic concepts and relations between them.
- *Spatiotemporal queries*. Queries are related to spatial and temporal locations of objects and video segments within the video.
- *Composite queries*. These queries contain any combination of other simple queries. The user composes the query by putting together image and video segments and specifying their properties, then asks the system to retrieve similar ones from the database.

We developed BilVideo-7 as a comprehensive, MPEG-7-compatible system to support such multimodal queries in a video indexing and retrieval framework. The name BilVideo-7 derives from BilVideo, which was a prototype video database system that supported keyword-based spatiotemporal queries using a knowledge base and a Prolog inference engine.[2]

To create BilVideo-7, we designed an MPEG-7 profile for video representation that enables detailed queries on videos, and used our

1070-986X/10/$26.00 © 2010 IEEE

Published by the IEEE Computer Society

MPEG-7-compatible video feature extraction and annotation tool to obtain MPEG-7-compatible video representations. The BilVideo-7's visual query interface, which is designed to help formulate complex multimodal queries, supports a comprehensive set of MPEG-7 descriptors. The queries for this system are processed on a multithreaded query-processing server with a multimodal query-processing architecture suitable for parallelization.

MPEG-7 foundation

MPEG-7 is an ISO/IEC standard developed by Moving Picture Experts Group (MPEG), 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 the Multimedia Content Description Interface.

MPEG-7 offers a comprehensive set of audiovisual description tools in the form of descriptors and description schemes that describe the multimedia data, forming a common basis for applications. The Description Definition Language is based on W3C XML with some MPEG-7-specific extensions, such as vectors and matrices. 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[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 descriptions but it doesn't specify how the description is produced. It's up to the developers of MPEG-7-compatible applications to determine 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 descriptors, which include Color Structure Descriptor (CSD), Scalable Color Descriptor (SCD), Dominant Color Descriptor (DCD), Color Layout Descriptor (CLD), Face Recognition Descriptor (FRD), and Group-of-Frame (GoF) or Group-of-Picture (GoP) descriptor.
- Texture descriptors, which include Edge Histogram Descriptor (EHD), Homogeneous Texture Descriptor (HTD), and Texture Browsing Descriptor.
- Shape descriptors, which include Contour Shape Descriptor and Region Shape Descriptor (RSD).
- Motion descriptors, which include Motion Activity, Motion Trajectory (MTD), Camera Motion, and Parametric Motion Descriptors.
- Localization descriptors, which include Region Locator and Spatiotemporal Locator Descriptors.

In MPEG-7, the semantic content of multimedia (for example, objects, events, and concepts) can be described by text annotation (free text, keyword, and structured) and semantic-entity and semantic-relation tools. Free-text annotations describe the content using unstructured natural language text (for example, 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 (for example, Barack Obama, visit, Turkey, and 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 (for example, who: Barack Obama, what action: visit, where: Turkey, and 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 (for example, agent, agentOf, patient, patientOf, result, resultOf, similar, opposite, user, userOf, location, locationOf, time, and timeOf) or by nonnormative relations.

MPEG-7's semantic tools provide methods to create brief or extensive semantic descriptions of multimedia content. Some of the descriptions can be obtained automatically, while most of them require manual labeling. For example, transcribed text obtained from automatic speech-recognition 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. Descriptions of semantic entities and relations between them, which cannot be obtained automatically with the current state-of-the-art technologies, require a considerable amount of manual work.

In 2007, MPEG adopted the MPEG Query Format[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:

- Input query format defines the syntax of query messages sent by a client to the server and supports different query by free text, query by description, query by XQuery, spatial query, temporal query, and so on.
- Output query format specifies the structure of the result set to be returned.
- Query management tools are used to search and choose the desired services for retrieval.

See the ''MPEG-7 Compatible Systems'' sidebar for related MPEG-7-compatible multimedia systems.

MPEG-7-compatible video representation

The first step in constructing an MPEG-7-compatible video-management system is to decide what query types to support and then to design an MPEG-7 profile. Representing video is crucial because 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 result in longer response time during retrieval. Keeping these factors in mind, we adapted our MPEG-7 profile from the one described in Bailer and Schallauer[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 our system can support a wide range of queries. First, audio and visual data are separated (using media-source decomposition[1]). Then, visual content is hierarchically decomposed into smaller structural and semantic units. Figure 1 (on page 66) shows an example of video decomposition according to this profile. First, the video is partitioned into non-overlapping video segments called *shots*, each having a start time and duration, with semantic annotation to describe the objects or events, and visual descriptors such as motion and group of frame or group of picture.

Next, the method performs temporal decomposition on the shots. The background content of the shots doesn't change much, especially if the camera is not moving. This static content can be represented by a single 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 annotation, and a set of visual descriptors. The visual descriptors are extracted from the frame as a whole. In addition, each keyframe is decomposed into a set of still regions (spatiotemporal decomposition) to keep more detailed region-based information in the form of spatial location by the minimum bounding rectangles (MBRs) of the region, semantic annotation, and region-based visual descriptors.

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. For example, faces are represented by moving regions, having an additional visual descriptor: the FRD.

Because the position, shape, motion and visual appearance of the salient objects might change throughout the shot, descriptors sampled at appropriate times should be stored. The trajectory of an object is represented by

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. One approach proposes an MPEG-7 profile for detailed description of audiovisual content that can be used in a broad range of applications.[1]

An MPEG-7-compatible database system extension to Oracle Database Management System is proposed in *MPEG-7 Multimedia Database System*.[2] The system is demonstrated by audio and image retrieval applications. Other research has proposed algorithms for the automatic generation of three MPEG-7 description schemes: video table of contents, for active video browsing; summary, to enable the direct use of metadata annotation of the producer; and still image, to allow interactive, content-based image retrieval.[3] Tseng et al. address the issues associated with designing a video personalization and summarization system in heterogeneous environments using MPEG-7 and MPEG-21.[4]

IBM's VideoAnnEx Annotation Tool (see http://www.research.ibm.com/VideoAnnEx) 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 the user might provide. The tool is limited to concept annotation and cannot extract low-level MPEG-7 descriptors from the video. The M-OntoMat-Annotizer software aims at linking low-level MPEG-7 visual descriptions to conventional Semantic Web ontologies and annotations.[5] The visual descriptors are expressed in the Resource Description Framework. The IFinder system was developed to produce limited MPEG-7 representation from audio and video by speech processing, keyframe extraction, and face detection.[6] ERIC7 is a software testbed that implements content-based image retrieval using image-based MPEG-7 color, texture, and shape descriptors.[7] Caliph and Emir are MPEG-7 based Java prototypes for digital photo and image annotation and retrieval, supporting graph-like annotations for semantic metadata and content-based image retrieval using MPEG-7 descriptors.[8] Cao et al. describe a middleware solution to access a bundle of MPEG-7 based multimedia services from mobile devices.[9]

These MPEG-7 compatible systems have two major problems. 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 limits the range of queries. Next, 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 and using a multimodal query processing and bottom-up subquery result fusion architecture to support complex multimodal queries (such as composite queries) with a comprehensive set of MPEG-7 descriptors.

References

1. W. Bailer and P. Schallauer, ''Detailed Audiovisual Profile: Enabling Interoperability Between MPEG-7 Based Systems,'' *Proc. 12th Int'l Multi-Media Modelling Conf.,* IEEE Press, 2006, pp. 217-224.
2. M. Döller and H. Kosch, ''The MPEG-7 Multimedia Database System (MPEG-7 MMDB),'' *The J. Systems and Software,* vol. 81, no. 9, 2008, pp. 1559-1580.
3. Y. Rui, ''MPEG-7 Enhanced Ubi-Multimedia Access–Convergence of User Experience and Technology,'' *Proc. 1st IEEE Int'l Conf. Ubi-Media Computing,* IEEE Press, 2008, pp. 177-183.
4. B. Tseng, C.-Y. Lin, and J. Smith, ''Using MPEG-7 and MPEG-21 for Personalizing Video,'' *IEEE Multimedia,* vol. 11, no. 1, 2004, pp. 42-52.
5. K. Petridis et al., ''M-OntoMat-Annotizer: Image Annotation Linking Ontologies and Multimedia Low-Level Features,'' LNCS 4253, Springer, 2006, pp. 633-640.
6. J. Löffler et al., ''IFinder: An MPEG-7-Based Retrieval System for Distributed Multimedia Content,'' *Proc. 10th ACM Int'l Conf. Multimedia,* ACM Press, 2002, pp. 431-435.
7. L. Gagnon, S. Foucher, and V. Gouaillier, ''ERIC7: An Experimental Tool for Content-Based Image Encoding and Retrieval Under the MPEG-7 Standard,'' *Proc. Winter Int'l Symp. Information and Communication Technologies,* 2004, ACM Press, pp. 1-6.
8. M. Lux, J. Becker, and H. Krottmaier, ''Caliph&Emir: Semantic Annotation and Retrieval in Personal Digital Photo Libraries,'' *Proc. CAiSE Forum at 15th Conf. Advanced Information Systems Engineering,* Springer, 2003, pp. 85-89.
9. Y. Cao et al., ''Mobile Access to MPEG-7 Based Multimedia Services,'' *Proc. 10th Int'l Conf. Mobile Data Management,* ACM Press, 2009, pp. 102-111.

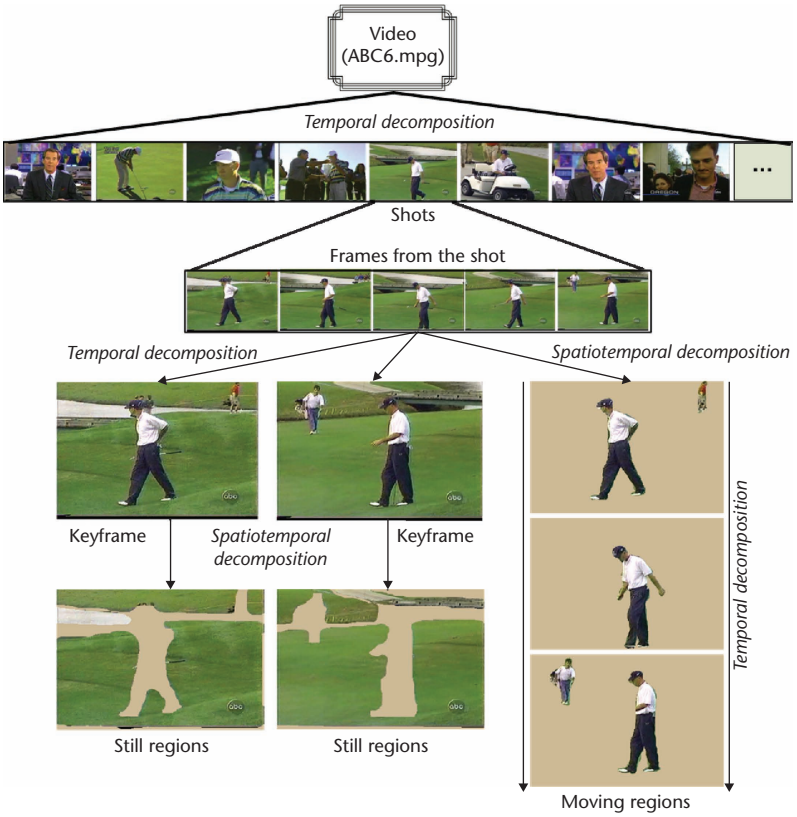


Figure 1. Decomposition of a video according to the MPEG-7 profile used in BilVideo-7. Low-level color, texture, and shape descriptors of the still and moving regions are extracted from the selected arbitrarily shaped regions, but the locations of the regions are represented by their minimum bounding rectangles.

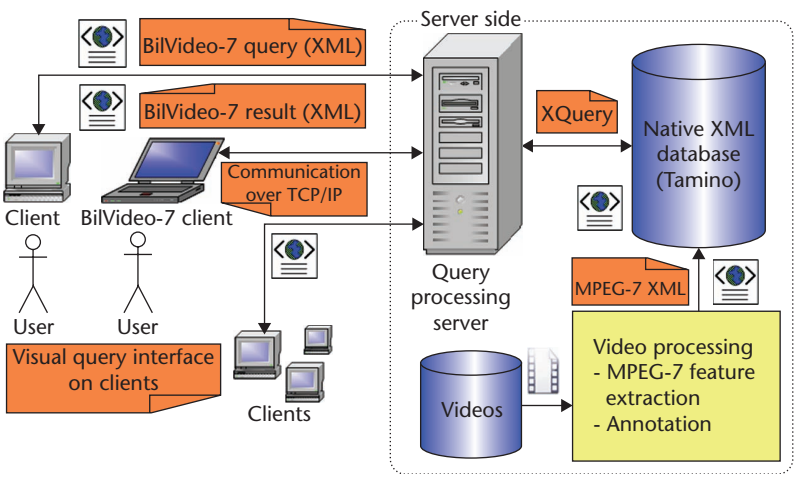


Figure 2. BilVideo-7's client–server architecture.

the MTD. The MBRs and visual descriptors of the object throughout the shot are stored by temporally decomposing the object into still regions. In this article, we refer to shots, keyframes, still regions, and moving regions as *video segments*.

System architecture

BilVideo-7 has a client–server architecture (see Figure 2). Users formulate queries on BilVideo-7 clients, which communicate with the BilVideo-7 query-processing server using an XML-based query language (see http://www.cs.bilkent.edu.tr/bilmdg/bilvideo-7) 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.

Feature extraction and annotation

MPEG-7 video representations are obtained using the MPEG-7-compatible video feature extraction and annotation tool (see Figure 3a). In the figure, the current video frame is shown at the top left, the latest processed frame in the bottom left, the latest selected region in the top right, and the selected moving regions along with their trajectories in 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, and localization) for the selected video segments are computed by the tool, using an MPEG-7 feature-extraction library adapted from MPEG-7 eXperimentation Model (XM) Reference Software.[3] The semantic content is described by text annotations (free text, keyword, and structured annotation).

The output is saved as an MPEG-compatible XML file for each video. We use a native XML database, Tamino (see http://www.softwareag.com/Corporate/products/wm/tamino/default.asp), to store the MPEG-7 representations. Native XML databases use the XML data model directly. Thus, it's more convenient and natural to use a native XML database because no mapping and conversion to other data models is required and it's easy to set up the database using the publicly available MPEG-7 schema.

Visual query interface

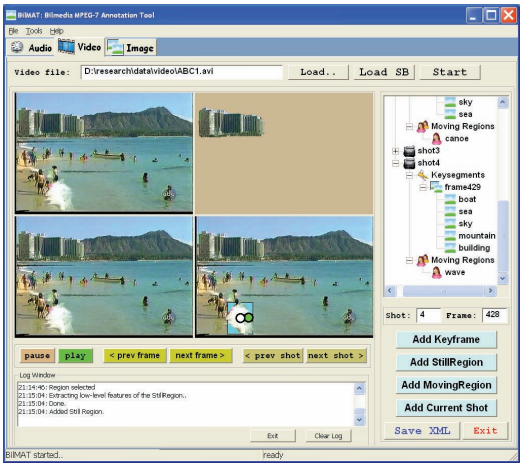
Users formulate queries on BilVideo-7 clients' visual-query interface, which consists of several tabs, each for a different type of query with a comprehensive set of descriptors and query options. As shown in Figure 3b, the query-formulation tabs are on the left, the query result list is displayed at the top right, the query results are at the bottom right, and the messages are 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 the BilVideo-7 query format in XML and sent to the BilVideo-7 query-processing server.

The video table of contents 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 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, moving regions, temporal location, and the semantic concepts with which the the videos are annotated.

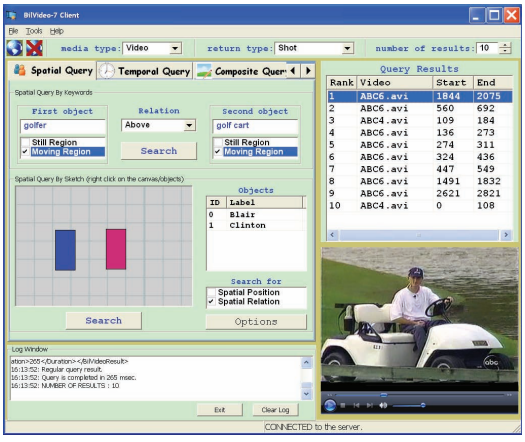
The textual query interface enables the user to formulate high-level semantic queries by entering keywords and specifying the type of video segment and annotation to be searched. The color, texture, and shape query interface is used for querying video segments by MPEG-7 color, texture, and shape descriptors. The input media can be a video segment, a whole image, or an image region. To provide this query functionality, 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. the BilVideo-7 clients include the MPEG-7 feature-extraction module to extract these descriptors. In addition to these query options, the user can specify the type of video segments to search, and also pick the weights and thresholds for each type of descriptor.

The motion-query interface is for the formulation of Motion Activity and Motion Trajectory queries. Trajectory points are entered using the mouse. The user can specify keywords for the moving region on which the trajectory query will be performed. The 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, and so on) or by sketching the MBR of objects using the mouse, and if desired, giving labels to them. It's possible to query objects on the basis of location, spatial relations, or both. The temporal-query interface is similar to the spatial query interface; this time, the user specifies temporal relations between video segments either by selecting from a predefined list (before, after, during, and so on) or by sketching the temporal positions of the segments using the mouse.

The composite-query interface is the most powerful query interface and enables the user to formulate complex queries. The query is composed by putting together any number of shots, keyframes, still regions, and moving



(a)



(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.

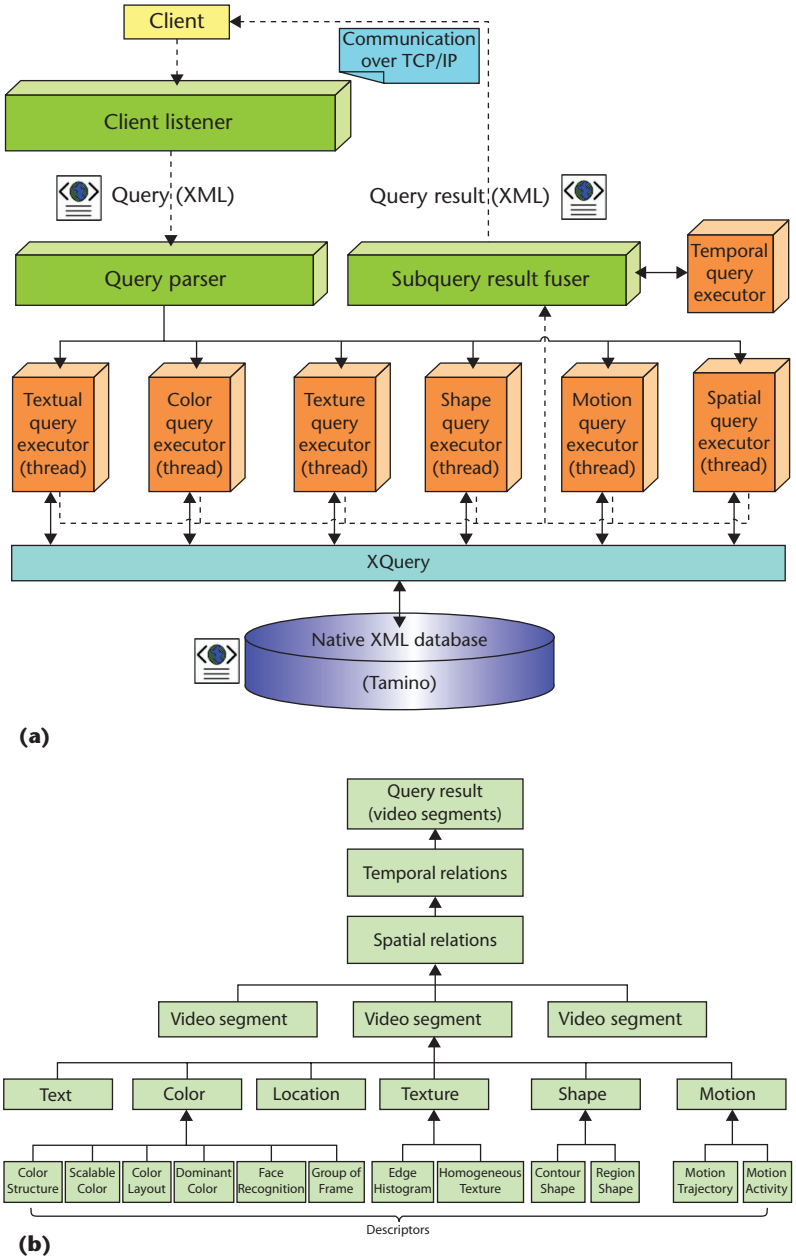


Figure 4. (a) The framework of the Query Processing Server. Each type of subquery is executed in a separate thread. (b) Subquery results are fused in a bottom-up manner.

regions, then specifying their properties as text-based semantic annotations, visual descriptors, locations, and 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.

The XQuery interface is best suited to experienced users who can write XQueries to search in the database. This is the most flexible interface in the system, and allows users to specify a wide range of queries.

Query processing

Query processing is done on the query-processing server, which is a multithreaded 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-7 query format and receive query results in XML-based BilVideo-7 result format, which contains a list of video segments (video name, start time, and end time) in ranked order. The current version of BilVideo-7 doesn't support MPEG Query Format query language because it's not possible to formulate some of the BilVideo-7 queries in MPQF (for example, spatial queries by location).

Multithreaded query execution

Each incoming query is parsed into subqueries and executed in a multithreaded fashion, with one thread for each type of subquery, as shown in Figure 4a. Queries with the same subquery type are accumulated in a queue and executed on a first-in-first-out basis. For example, subqueries for color descriptors (such as CSD, SCD, and DCD) are added to the end of queue of the 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 (for example, 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 because 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 (for example, CSD for keyframes) and computes their distances to the query.

The distance measures suggested by MPEG-7 are implemented in MPEG-7 XM Reference Software but they are not normative. The evaluation of the distance measures for a set of MPEG-7 descriptors, presented elsewhere,[6] shows that although there are better distance measures (for example, pattern difference or Meehl index), the MPEG-7 recommendations

are among the best. Therefore, we adapted the distance measures from the XM software and briefly describe them here. In our computations, Q refers to a descriptor in the query, D to a descriptor in the database, and d is the computed distance between the descriptors.

We use L_1–norm to compute the distance between the CSD, SCD, GoF/GoP, and RSD.

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

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

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

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 EHDs Q and D is computed by adapting the L_1–norm as

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

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 semiglobal edge histograms.

The distance between two HTDs 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 FRDs 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 ith 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, we use Euclidean distance. For spatial-position queries, we use Euclidean distance between the center points of objects' MBRs. 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 (for example, for a 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 $s(Q, D) = 1 - d(Q, D)/d_{max}$, $0 \leq s(Q, D) \leq 1.0$; where $d(Q, D)$ is the distance between descriptors Q and D, and d_{max} is the maximum possible distance for the type of descriptor in the computation. We compute the maximum distance for each descriptor by taking the maximum distance from a large set of descriptors extracted from video segments.

Spatial locations of still and moving regions are stored in the database by their MBRs, without any preprocessing to extract and store the spatial relations between them. Therefore, we compute spatial similarity between regions at query-execution time, which is computationally more expensive but provides a more flexible and accurate matching for spatial-position and spatial-relation queries.

For each still and moving region in the query, queries related to the properties of the region are executed as described previously. Then the resulting video segments undergo spatial query processing to compute the spatial

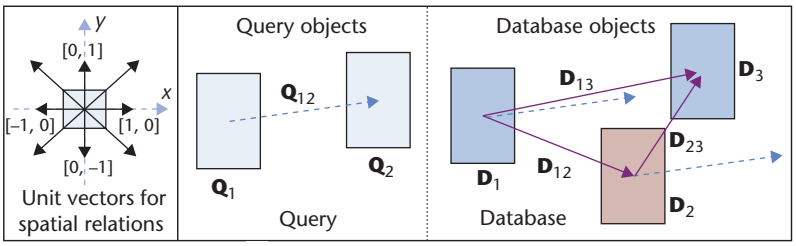


Figure 5. Spatial query processing by vector dot product between the vectors connecting centers of objects' minimum bounding rectangles.

similarities between them. We use the spatial similarity matching approach described elsewhere[7] because of its efficiency and robustness. First, we compute the vectors connecting the center points of objects' MBRs, $\mathbf{Q_{xy}}$ and $\mathbf{D_{ij}}$, as shown in Figure 5. Then, we compute the pairwise spatial similarity by the cosine of the angle between the vectors $\mathbf{Q_{xy}}$ and $\mathbf{D_{ij}}$, using vector dot product

$$d(\mathbf{Q_{xy}}, \mathbf{D_{ij}}) = \cos\theta = \frac{\mathbf{Q_{xy}} \cdot \mathbf{D_{ij}}}{|\mathbf{Q_{xy}}||\mathbf{D_{ij}}|}, \quad 0 \le \theta \le \pi,$$

$$-1 \le d(\mathbf{Q_{xy}}, \mathbf{D_{ij}}) \le +1$$

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, and so on) are executed in the same way, by converting each spatial-relation query to a unit vector (see Figure 5). For instance, Q_x *right* Q_y (Q_x is to the right of Q_y) query is converted to a query vector $\mathbf{Q_{xy}} = [-1, 0]$, from Q_x to Q_y. Multiple MBRs are stored in the database for moving regions to keep track of their locations. We compute the spatial similarity for all the MBRs and take the maximum similarity value as the final similarity. We execute temporal queries, if any, 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 because they must co-appear on a scene for a certain time in the video to satisfy the spatial relations.

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 fusion is done in a bottom-up manner, as shown in Figure 4b. Each node in the tree has an associated weight and threshold, which can be specified by the user during query formulation. We compute the similarity at each node as the weighted average of the similarities of its children. The fusion process continues upward in the tree until we obtain the final query result.

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 five subqueries (CSD, DCD, EHD, HTD, and semantic), executes each, and produces five lists of keyframes from the 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:

$$s_{i,color} = \frac{w_{keyframe,CSD}s_{i,CSD} + w_{keyframe,DCD}s_{i,DCD}}{w_{keyframe,CSD} + w_{keyframe,DCD}}$$

$$s_{i,texture} = \frac{w_{keyframe,EHD}s_{i,EHD} + w_{keyframe,HTD}s_{i,HTD}}{w_{keyframe,EHD} + w_{keyframe,HTD}}$$

where $s_{i,color}$ is the color similarity for the ith 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 three lists of keyframes having similarity values for color, texture, and semantics (text). We fuse these three 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 and by sorting the list in descending order of similarity.

Experimental results

The system is implemented in C++ and uses the Open Source Computer Vision Library (see http://opencvlibrary.sourceforge.net) to handle

(a)

(b)

(c)

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

the image and video data, and wxWidgets (see http://www.wxwidgets.org) for the graphical user interfaces. We performed queries on a video data set consisting of 14 video sequences with 25,000 frames from the Trecvid 2004 and 2008 data sets,[8] which include news, documentary, educational, and archival videos. The query result is a list of shots in ranked order, each shown as a representative keyframe in Figure 6.

Figure 6a shows two spatial queries. 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

Table 1. Query-execution times. The query-processing server and Tamino XML database are installed on a notebook PC with Intel Core 2 dual-core 2-GHz processors and 2 Gbytes of RAM, running Windows XP.

Query type	Description (Segments and descriptors)	Execution time (seconds)
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 and texture query	Keyframe (CSD and HTD)	0.172
Shape query	Moving region (RSD)	0.156
Spatial query	Text-based, two still regions	0.172
Spatial query	Text-based, two moving regions	0.187
Spatial query	Sketch-based, two moving regions	0.187
Composite query in Figure 6c, top row	Keyframe (DCD and keyword), moving region (CSD, RSD, and keyword)	0.438
Composite query in Figure 6c, bottom row	Two still regions (CSD and EHD), moving region (keyword)	0.391

that exactly match the spatial query condition. The fourth result contains a golfer but no golf cart; the spatial condition is not satisfied, so its rank is lower than the first three.

The second query on the bottom row of Figure 6a, ''Clinton left Blair,'' is sketch-based. The spatial-query condition is satisfied exactly in the first two video segments returned, while it's not satisfied in the last two, even though 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.

Figure 6b shows two low-level queries. In the image-based query (top row), the query image is represented by CSDs and DCDs and searched in keyframes. In the region-based query (bottom row), the query region is represented by CSDs and RSDs, and searched in moving regions. Both query results are satisfactory considering the types of descriptors used.

The three queries shown in Figure 6c are composite queries, in which high-level semantics in the form of keyword annotations and low-level descriptors (DCD, CSD, EHD, and RSD) are used together to describe the query video segments. In the first composite query, the keyframe is represented by the DCD and golf green. The moving region is represented by the CSD, RSD, and golfer. In the second query, two still regions at the top and at the bottom of Figure 6c are represented by the CSD and EHD. 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 flexible and powerful, enabling the user to formulate 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), sizes of the descriptors, and complexity of the matching algorithm (distance metric). Queries involving low-level descriptors take longer to execute in comparison to text-based queries because the distance computations between the low-level descriptors are computationally more expensive. The multithreaded query-processing architecture provides some degree of parallelism and shortens the query-execution times when the subqueries are executed in separate threads.

Conclusions and future work

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. However, the system's retrieval performance depends very much on the MPEG-7 descriptors and the distance measures used. As for our future work, we plan to investigate distance measures other than the ones recommended by MPEG-7.[6]

Currently, 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 suffers from human subjectivity. This processing 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.

In future versions of BilVideo-7, we plan to add support for representation and querying of audio and image data. The BilVideo-7 multimodal query-processing architecture makes it easy to add new descriptors for new modalities such as audio descriptors. Images could be considered to be a special case of keyframes that are decomposed into still regions, and hence can be supported easily in BilVideo-7. **MM**

Acknowledgments

We thank Rana Nelson for proofreading this manuscript.

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 et al., ''BilVideo: Design and Implementation of a Video Database Management System,'' *Multimedia Tools and Applications,* vol. 27, no. 1, 2005, pp. 79-104.
3. *Information Technology–Multimedia Content Description Interface–Part 6: Reference Software,* ISO/IEC 15938-6:2003, 2009; http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html.
4. *Information Technology–Multimedia Content Description Interface–Part 12: Query Format,* ISO/IEC FCD 15938-12:2007, 2007; 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,'' *Proc. 12th Int'l Multi-Media Modelling Conf.,* IEEE Press, 2006, pp. 217-224.
6. H. Eidenberger, ''Distance Measures for MPEG-7-Based Retrieval,'' *Proce 5th ACM SIGMM Int'l Workshop Multimedia Information Retrieval,* ACM Press, 2003, pp. 130-137.
7. J.Z. Li and M.T. Ozsu, ''Stars: A Spatial Attributes Retrieval System for Images and Videos,'' *Proc. 4th Int'l Conf. Multimedia Modeling,* Springer, 1997, pp. 69-84.
8. A.F. Smeaton, P. Over, and W. Kraaij, ''Evaluation Campaigns and Trecvid,'' *Proc. 8th ACM Int'l Workshop Multimedia Information Retrieval,* ACM Press, 2006, pp. 321-330.

Muhammet Başştan is a PhD 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 and video processing, saliency, segmentation, and annotation. Başştan has an MS in electronics engineering and computer science from Sabanci University, Istanbul, Turkey. Contact him at bastan@cs.bilkent.edu.tr.

Hayati Çam (1983-2009) was a software engineer at Siemens, Ankara, Turkey. His research interests included multimedia databases and computer vision. Çam had an MS from the Department Computer Engineering at Bilkent University.

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. Güdükbay has a PhD in computer engineering and information science from Bilkent University. He is a senior member of IEEE and ACM. Contact him at gudukbay@cs.bilkent.edu.tr.

Özgür Ulusoy is a 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. Ulusoy has a PhD in computer science from University of Illinois at Urbana-Champaign. He is a member of IEEE and ACM. Contact him at oulusoy@cs.bilkent.edu.tr.