Accommodating Hybrid Retrieval in a Comprehensive Video Database Management System

Shermann S.M. Chan       Qing Li*       Yi Wu      Yueting Zhuang

Department of Computer Science
City University of Hong Kong, China

Abstract

A comprehensive video retrieval system should be able to accommodate and utilize various (complementary) description data in facilitating effective retrieval. In this paper, we advocate a hybrid retrieval approach by integrating a query-based (database) mechanism with content-based retrieval (CBR) functions. We describe the VideoMAP+ architecture, discuss issues related to developing such a comprehensive video database management system, and its specific language mechanism (CAROL/ST with CBR) which provides an improved expressive power than what pure query-based or CBR methods currently offer. We also describe an experimental prototype being developed based on a commercial object-oriented toolkit using VC++ and Java.

1. Introduction

A current important trend in multimedia information management is towards web-based/enabled multimedia search and management systems. Video is a rich and colorful media widely used in many of our daily life applications like education, entertainment, news spreading, etc. Digital videos have diverse sources of origin such as cassette recorder, tape recorder, home video camera, VCD and Internet. Expressiveness of video documents decides their dominant position in the next-generation multimedia information systems. Unlike traditional / static types of data, digital video can provide more effective dissemination of information for its rich content. Collectively, a (digital) video can have several information descriptors: (1) metadata - the actual video frame stream, including its encoding scheme and frame rate; (2) media data - the information about the characteristic of video content, such as visual feature, scene structure and spatio-temporal feature; (3) semantic data - the text annotation relevant to the content of video, obtaining by manual or automatic understanding.

Video metadata is created independently from how its contents are described and how its database structure is organized later. It is thus natural to define “video” and other meaningful constructs such as “scene”, “frame” as objects corresponding to their respective inherent semantic and visual contents. Meaningful video scenes are identified and associated with their description data incrementally. But the gap
between the user realization and video content remains a big problem. Depending on the user’s viewpoint, the same video/scene may be given different descriptions. It is extremely difficult (if not impossible) to describe the whole contents of a video, especially due to the visual content.

1.1 Background of Research

Over the last couple of years we have been working on developing a generic video management and application processing (VideoMAP) framework [CL99a, CL99b, LLS00]. A central component of VideoMAP is a query-based video retrieval mechanism called CAROL/ST, which supports spatio-temporal queries [CL99a, CL99b].

While the original CAROL/ST has contributed on working with video semantic data based on an extended object oriented approach, little support has been provided to support video retrieval using visual features. To come up with a more effective video retrieval system, we have been making extensions to the VideoMAP framework, and particularly the CAROL/ST mechanism to furnish a hybrid approach [CWLZ01]. In this paper we thus present VideoMAP+, a successor of VideoMAP, which has an extended capability of supporting the hybrid approach to video retrieval through integrating the query-based (database) approach with the CBR paradigm.

1.2 Paper Contribution and Organization

In order to develop an effective video retrieval system, one should go beyond the traditional query-based or purely content-based retrieval (CBR) paradigm. Our standpoint is that videos are multi-faceted data objects, and an effective retrieval system should be able to accommodate all of the complementary information descriptions for retrieving videos. In this paper, we discuss the main issues involved in developing such a comprehensive video database management system supporting hybrid retrieval.

The rest of our paper is organized as follows. In next section we review some related on video processing and database management. Section 3 is devoted to the introduction of the hybrid approach to video retrieval undertaking by VideoMAP+; the CBR and query-based retrieval methods are elaborated and their integration into a single language framework is presented. In section 4, we describe an experimental prototype system which we have been building, by highlighting on the main user interface facilities; sample queries are also given to illustrate the expressive power of this mechanism. Finally, we conclude the paper and offer further research directions in section 5.

2. Related Work

There has been significant interests and considerable amount of research in developing management systems for video databases in recent years. Here we review some existing work with an attempt to compare and contrast different approaches of modeling and managing video data. While there have been several research projects on video databases initiated, the following are what we regard as representative ones which
support either content-based search or annotation/query-based retrieval techniques in their models and systems. Besides, some existing work on hybrid retrieval of image and video is also discussed below.

### 2.1 Content-based retrieval querying

A lot of existing work deals with **content-based retrieval**; a good survey on this topic is given in [YI99]. Dimitrova and Golshani [DG94] developed Rx to retrieve video data based on the trajectory of objects. They modeled video with multi-resolution semantic hierarchy, but the object descriptor in this model is placed at a layer lower than image features. Hence it lacks the independency of physical video data and conceptual metadata, and can handle only the semantics of moving trajectory. They have also developed a visual language VEVA for content-based video retrieval [GD98], which can formulate queries to access semantic information contained in digital video, and motion information as well. At IBM Almaden research center, a system called QBIC [Fli+95] was developed to support image/video data access using color, shape, texture and various properties of raw data. However, QBIC does not have a special abstraction model and can only classify raw data into physical data or feature data. Smoliar and Zhang [SZ94] proposed a video indexing and retrieval approach by focusing on the content analysis process. They emphasized on the need of a good model to represent analyzed semantics efficiently, but did not provide an efficient mechanism to manage and query video data. Swets, Pathak and Weng [SPW98] developed an image database system which supports query-by-image-example, in addition to simple alphanumeric queries which involve basic field name/value matching.

### 2.2 Video modeling and querying

There has also been some work on **video modeling and querying**. Chua and Ruan [CR95] presented a two-layered (shot layer and scene layer) conceptual model, in which the shot layer contains a collection of video frames and the scene layer is used to model the domain knowledge and contextual information of the videos. This separation, however, is based on the granularity of concept, not on different properties of data. There are also few facilities provided for declarative and associative search. Gibbs et al. [GBT94] proposed a model for the interpretation, derivation, and temporal composition of media objects. A media object is defined as a timed audio/video/audio-video stream. While this model provides good support for the editing and presentation of audio-video content, it lacks an easy retrieval facility to audio-video streams based on inherent items of interests or metadata. Hjelsvold and Midstraum [HM94] presented a video data model and a query language extension. They adopted the stratification approach and proposed two levels of abstraction as the entire video document and the individual frames. They introduced a logical concept named **video stream**, which allows users to annotate a description on any part of the video data. However, this annotation scheme depends on the physical structure of a video stream, and hence lacks data independency. Oomoto and Tanaka [OT93] proposed an object-oriented video data model, OVID, in which they introduced the notion of **video objects** to facilitate the identification and the composition of meaningful features. This model is “type-weak” and offers a flexible framework for organizing a lot of video descriptonal data, but it does not provide an efficient structure of the desciptional data nor a clear separation of temporal data from descriptonal annotations. Schloss and Wynblatt [SW95] introduced a layered multimedia data model and provided repositories of reusable data shared...
among various applications. They separated the concept of multimedia data into data definition, presentation and temporal structure, but they do not provide conceptual structure for efficient query processing. Jiang and Elmagarmid [JE98] presented a video database system called WVTDB which supports semantic modeling and content-based search on the world wide web (WWW). Lee et. al. [Lee+99] developed an icon-based, graphical query language GVISUAL which allows a user to specify temporal queries using iconic/graphical representations, but no support facility for content-based retrieval is provided.

2.3 Hybrid image and video retrieval

In addition, some work on hybrid query retrieval of image and video was surveyed and reviewed in [YM98]. Chabot [OS95] is a picture retrieval system and its object identification is based on color analysis semi automatically. It uses both keywords and image features for retrieval. In [CH92, GWJ91], the authors gave an idea of building a hierarchy of image representations from raw image data to objects and relations at the user semantic level. Chang and Hsu [CH92] analyzed raw image data in terms of their geometric patterns, scenes with semantics, and some meaningful entities. The overall information could be utilized in spatial reasoning and image retrieval. Gupta, Weymouth and Jain [GWJ91] have developed a VIMSYS model for querying a pictorial database. SEMCOG [LCH97] is an object-based image retrieval system which integrated semantic and cognition-based information for retrieval. Besides hybrid image retrieval, there have been some hybrid video querying systems [ZLSW95, ZWLS95, CA96]. Zhang et. al. [ZLSW95, ZWLS95] parsed and decomposed raw video data into shots and scenes automatically; while JACOB [CA96] used camera operations to split a video into shots. They both use keywords and low-level image features for video retrieval and browsing.

2.4 Relevance to MPEG-7

MPEG-7 standard, a means of attaching metadata to multimedia content, is often called “Multimedia Content Description Interface”. It aims at providing a rich set of audiovisual description tools to describe multimedia content data, which will support some degree of interpretation of the information’s meaning. It is intended to describe audiovisual information regardless of storage, coding, display, transmission, medium, or technology. Audiovisual data content may include still pictures, graphics, 3D models, audio, speech, video, and composition information about how these elements are combined in a multimedia presentation. Special cases of these general data types may include facial expressions and personal characteristics. MPEG-7 work can be separated into three parts: Descriptors, Description Schemes, and a Description Definition Language. Descriptors are the representations of low-level features. Description Schemes are structured combinations of Descriptors, and they can be used to form a richer expression of a higher-level concept. The Description Definition Language is the language that allows the creation of new Description Schemes and Descriptors. It also allows the extension and modification of existing Description Schemes.
There are many MPEG-7-related projects being undertaken within commercial enterprises, particularly broadcasting and digital imaging companies.\(^9\) [HARMONY] is a three-way International Digital Libraries Initiative project between Cornell University, the Distributed Systems Technology Centre, and the University of Bristol’s Institute for Learning and Research Technology. Its objective is to develop a framework to deal with the challenge of describing networked collections of highly complex and mixed-media digital objects. The research draws together works on the RDF, XML, Dublin Core, MPEG-7 and INDECS standards, and focuses on the problem of allowing multiple communities of expertise (e.g., library, education, rights management) to define overlapping descriptive vocabularies for annotating multimedia content.

DICEMAN (Distributed Internet Content Exchange with MPEG-7 and Agent Negotiations) project is an EC-funded project between Teltec Ireland DCU, CSELT (Italy), IBM (Germany), INA (France), IST (Portugal), KPN Research (Netherlands), Riverland (Britain) and UPC (Spain) [DICEMAN]. Its objective is to develop an end-to-end chain for indexing, storage, search and trading of digital audiovisual content. The technical aspects of this project are mainly: MPEG-7 indexing through a Content Provider's Application (COPA); the use of Foundation for Intelligent Physical Agents (FIPA) to search and locate the best content; and support for electronic commerce and rights management.

The A4SM project, which is based on GMD's IPSI (Integrated Publication and Information Systems Institute), is currently researching the application of IT support to all stages of the video production process [IPSI]. The purpose is to seamlessly integrate an IT support framework into the production process, i.e., pre-production (e.g., script development, story boarding, etc.), production (e.g., collection of media-data by using an MPEG-2/7 camera, etc.), and the post-production (support of non-linear editing). In collaboration with TV-reporters, cameramen and editors they have designed an MPEG-7 camera in combination with a mobile annotation device for the reporter, and a mobile editing suite suitable for the generation of news-clips.

Overall, the MPEG-7 standard and its related projects concentrate on content description and metadata attachment to multimedia data. Few facilities and little support have been provided by them in terms of video query formulation, processing, and retrieval, which are exactly the main theme of this paper.

3. Hybrid Approach to Video Retrieval

In a Video Database Management system (VDBMS), there exists an important need for efficient retrieval facility of the voluminous data. Accordingly, many ways are put forward. Content-based retrieval, which uses visual features such as color, texture and shapes, provides a direct and an intuitive approach for video data. But for too complex visual information and the limitation of computer vision technique, CBR itself is

\(^9\) MPEG-7 is able to actually support a broad range of applications, and can make the web as searchable for multimedia content. It will allow also fast and cost-effective usage of the underlying data, by enabling semi-automatic multimedia presentation and editing.
inadequate for video management. This is because many (semantic) features of video data cannot be extracted out from the video itself automatically; moreover, video objects may share annotations or descriptions. Consequently, it is necessary and cost-effective to complement content-based retrieval with declarative, efficient query-based retrieval in the video database system.

To address these problems, we have been developing an object-oriented VDBMS to complement content-based access with high level (declarative) query-based retrieval. As video objects by their nature are rich in spatial and temporal semantics, a versatile modeling mechanism is devised to support Spatio-Temporal reasoning. Associated language facilities are also being developed to accommodate a wide range of video queries and annotation activities. In this section, we will firstly review the main facilities of our original VideoMAP - an object oriented video database system supporting Spatio-Temporal reasoning, including its language CAROL/ST. Then we describe our approach to integrate the CBR mechanism into it to facilitate a hybrid approach for video retrieval, and detail the extended language CAROL/ST with CBR functionalities.

3.1 VideoMAP: Basic Features and Functionality

Our Video DBMS - VideoMAP has its architecture as shown in Figure 3.1. It shows (a) the process of video classification and segmentation from (a.1) to (a.3); (b) the process of adding features into CCM/ST, both general and spatio-temporal, and building Scene-order/Feature-index trees from (b.1) to (b.2); (c) the process of adding spatio-temporal features into CCM/ST by an Event/Action model from (c.1) to (c.2); and (d) the process of querying from (d.1) to (d.3). If the user makes an Event/Action query, the process of query is from (d.1), and then to-and-from (e), then to (d.2) and (d.3).

3.1.1 CCM/ST: an extended OO mechanism supporting spatio-temporal Semantics

Cluster is a conceptual video scene collection and it consists of attributes, methods, and dynamic groupings of video objects in database, within which each group of video objects is linked with one role. Since meaningful video sequences are identified and segmented by VCC [LLX99], conceptual video objects are created and associated with their describing data incrementally and dynamically. Similar to cluster, role is a feature/index of video scene and it consists of attributes and methods. A cluster can be used to represent a Scene of a video program, and a role can be employed to represent a Feature/index within the scene. Specifically, a Scene S is a dynamic object that has a tripartite form:
\[ S = \langle A, M, X \rangle \]

where \( A \) is a set of scene attributes, \( M \) is a set of scene methods, and \( X \) is a set of the Feature-Video associations:

\[ X = \{ \langle F_i : V_i : \{ ST_i \} \rangle | 1 \leq i \leq n \} \]

where \( F_i \) is a feature, and \( V_i \) is a set of video objects that carry the feature within the scene, and \( ST_i \) is a ST-Feature which is optionally included in the Feature-Video association. ST-Feature is used when an image object can be identified from the video objects and when an image object is represented by a ST-Feature which contains a set of “Position and Frame numbers”, which means there is a movement of the object concerned. A feature \( F_i \) can be described as follows:

\[ F_i = \langle A', M' \rangle \]

where \( A' \) and \( M' \) are a set of attributes and methods defined by the feature \( F_i \); these attributes and methods are applicable to the video players of the feature. In our Video DBMS, a feature is the same as a semantic descriptor (a glossary term) that is of interest to end-users.

**Supporting Scene Ordering**

For video applications, an important concept to support is Table of Contents (ToC) which describes the structural as well as sequencing relationships of video data within a video program. Within a Scene-Feature-Video association, the video objects linked by the feature should form a subset of the constituents included by the associated scene. In order to support ToC in CCM, the notion of scene is extended by defining a temporal scene \( S_t \) as follows:

\[ S_t = \langle A, M, X, T \rangle \]

where \( A \) and \( M \) denote attributes and methods defined by the scene \( S_t \) respectively, \( X \) is a set of the Feature-Video associations, and \( T \) is a template of showing the temporal ordering of the video objects in the scene \( S_t \).

**Supporting Time Dimension**

In CCM/ST, two kinds of time dimension in temporal databases: (valid-time and user-defined time) are chosen to be supported due to their applicability to video data. Valid-time concerns the time a fact was true in reality; User-defined time (such as birthdate) provides a new data type (e.g., Date-time) to the user [TCG+93]. In addition, in order to provide a more flexible way to users, CCM/ST is designed to support valid-time at both “Object level” and “Attribute level” [TCG+93], as shown in Table 1. Therefore, CCM/ST can store the past histories and future plans of scenes and features, and thus can provide different versions of a video program.

<table>
<thead>
<tr>
<th>Time dimension</th>
<th>Lifespan</th>
<th>Associated object</th>
<th>Associated object component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid-time</td>
<td>Object level</td>
<td>Scene / Feature</td>
<td>Attribute, Method, Feature-Video association, ToC</td>
</tr>
<tr>
<td>Valid-time</td>
<td>Attribute level</td>
<td>Scene</td>
<td>Attribute, Method</td>
</tr>
<tr>
<td>Valid-time</td>
<td>Attribute level</td>
<td>Feature</td>
<td>Attribute, Method</td>
</tr>
<tr>
<td>User-defined time</td>
<td>Attribute level</td>
<td>Scene / Feature</td>
<td>Attribute</td>
</tr>
</tbody>
</table>

Table 1. Time dimensions supported by CCM/ST

**Supporting temporal functions and operators**

There are several types of temporal functions and operators, e.g. interval specification functions, interval ordering functions, interval ordering operators, and interval comparison operators [SHC98, VTS98]. Some
temporal functions adopted in CCM/ST are shown in Table 2 and Table 3. Besides, the Interval ordering operators are extended from seven to nine operators as shown in Table 4. As the Interval comparison operators are used to compare the temporal relationships between two explicit or implicit time/frame intervals, the result of the comparison is either true or false. These operators are used on ToC and Valid-time, the same way as the Interval ordering operators.

<table>
<thead>
<tr>
<th>Function</th>
<th>Returns</th>
<th>Used On</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_INTERVAL()</td>
<td>Frame interval of video objects</td>
<td>ToC</td>
</tr>
<tr>
<td>VT_INTERVAL()</td>
<td>Valid-time interval of a temporal instance</td>
<td>Valid-time</td>
</tr>
<tr>
<td>VALID()</td>
<td>Current valid element</td>
<td>Valid-time</td>
</tr>
<tr>
<td>DURATION()</td>
<td>Duration of frame/valid-time interval</td>
<td>ToC, Valid-time</td>
</tr>
<tr>
<td>DATE()</td>
<td>Date instant object</td>
<td>Valid-time</td>
</tr>
<tr>
<td>INTERVAL()</td>
<td>Date interval object</td>
<td>Valid-time</td>
</tr>
</tbody>
</table>

**Table 2. Interval specification functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Returns</th>
<th>Used On</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST()</td>
<td>First element</td>
<td>ToC, Valid-time</td>
</tr>
<tr>
<td>LAST()</td>
<td>Last element</td>
<td>ToC, Valid-time</td>
</tr>
<tr>
<td>NTH()</td>
<td>The Nth position of element</td>
<td>ToC, Valid-time</td>
</tr>
<tr>
<td>PREV()</td>
<td>Previous element</td>
<td>ToC, Valid-time</td>
</tr>
<tr>
<td>NEXT()</td>
<td>Next element</td>
<td>ToC, Valid-time</td>
</tr>
</tbody>
</table>

**Table 3. Interval ordering functions**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Diagram</th>
<th>Used On</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEFORE</td>
<td>a BEFORE b</td>
<td><img src="BEFORE" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>AFTER</td>
<td>a AFTER b</td>
<td><img src="AFTER" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>EQUAL</td>
<td>a EQUAL b</td>
<td><img src="EQUAL" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>START</td>
<td>a START b</td>
<td><img src="START" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>END</td>
<td>a END b</td>
<td><img src="END" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>JUST-BEFORE</td>
<td>a JUST-BEFORE b</td>
<td><img src="JUST-BEFORE" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>JUST-AFTER</td>
<td>a JUST-AFTER b</td>
<td><img src="JUST-AFTER" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>INSIDE</td>
<td>a INSIDE b</td>
<td><img src="INSIDE" alt="Diagram" /></td>
<td>ToC</td>
</tr>
<tr>
<td>OVERLAP</td>
<td>a OVERLAP b</td>
<td><img src="OVERLAP" alt="Diagram" /></td>
<td>ToC</td>
</tr>
</tbody>
</table>

**Table 4. Interval ordering operators; a and b are frame intervals of video objects**

Supporting spatial functions and operators

In order to support spatial semantics of video data, the following spatial functions and operators have been introduced in CCM/ST.

(A) **Spatial functions**

Spatial functions = \{TOP(), BOTTOM(), LEFT(), RIGHT(), MIDDLE()\}

(B) **1-D spatial operators**

Spatial operators for horizontal dimension = \{LEFT, H_MIDDLE, RIGHT\}

Spatial operators for vertical dimension = \{TOP, V_MIDDLE, BOTTOM\}
(C) 2-D spatial operators

As in [NSN95], the 1-D operators can be used in pairs to form 2-D ones for identifying objects in both horizontal and vertical dimensions. Such 2-D operators can be used to specify queries such as “which object is at TOP_LEFT” within a specified Frame contained in a SCENE, etc.

3.1.2 CAROL/ST: a query language supporting spatial and temporal primitives

VideoMAP* has a query language (CAROL/ST) which is devised based on CCM/ST features. Five primitive video retrieval functions have been supported, namely:

- select scenes/features/videos by their attribute restrictions
- select video players (video objects) by scene’s restriction
- select scenes by their features’ restrictions
- select features by their scenes’ restrictions
- select scenes by their video players’ restrictions

In addition, CAROL/ST provides a set of expressive spatial and temporal primitives. These temporal and spatial operators are introduced based on CCM/ST model; the following query examples demonstrate how they work.

(A) Selection concerning Valid-time:

Example: Consider that a user wants to retrieve all the video objects about “SportsNews” before a given date (say, today). The query can be specified by the following CAROL/ST statement (x is a set of scene objects):

\[
\text{SELECT } x \text{ FROM SCENE } y \\
\text{WHERE } y \text{ HAS FEATURE } \text{SportsNews} \\
\text{AND VT_INTERVAL}(y) \text{ BEFORE CURRENT_DATE;}
\]

All video objects are extracted from the result of the above query, x, which contains the feature “SportsNews” and the valid-time of each of the retrieved scenes is before current date. Therefore, “histories” of scenes about SportsNews can be retrieved.

(B) Selection concerning Spatial/Temporal semantics:

Example: To specify a query which is to retrieve some video clips showing that Clinton is with Lewinsky for no less than 4 seconds, the following CAROL/ST statement can be formulated (x is a set of scene objects):

\[
\text{SELECT } x \text{ FROM SCENE } y \\
\text{WHERE } y \text{ HAS FEATURE } \text{Clinton AND } y \text{ HAS FEATURE Lewinsky} \\
\text{AND F_INTERVAL(Clinton) EQUAL F_INTERVAL(Lewinsky)} \\
\text{AND DURATION(F_INTERVAL(Clinton)) >120;}
\]

The query above retrieves a scene, which has features “Clinton” and “Lewinsky”, and the two features have the same interval of the same video object with more than 120 frames (about 4 seconds).

* x is selected from scene objects, therefore it is a set of scene objects.
3.1.3 Specification Language

As shown in Figure 3.1, VideoMAP has a specification language (or content/semantic description definition language) which can be used by expert users to annotate user-interested Event/Action semantics into CCM/ST. It is also based on the spatio-temporal reasoning, which can be embedded in the Feature Index Hierarchy. In VideoMAP, basic indices are first generated by the Video Classification Component (VCC), which are then grouped and linked by the CCM/ST. These indices are normally the key frames of the video objects. In order to extract and compare the image features of the video objects spatially, ST-Feature is introduced to CCM/ST to accommodate possible space/time-dependent annotations. The structure of the ST-Feature is as follow:

$$ST-Feature = \langle \{ \text{Position-array}, \text{Start-frame}, \text{End-frame} \} \rangle$$

where Position-array is the spatial representation of the image feature, and together with Start-frame and End-frame store the duration of the feature denoted by ST-Feature. Note that ST-Feature is used when an image object can be identified from the video objects. Besides static object, ST-Feature can represent a moving object too. Thus, if there are some features in the same scene, by using an Activity model shown in Figure 3.2, we can decide if some events and actions happened in the video object.

In Figure 3.2, the left-hand side is the architecture of the Activity Model and right-hand side is a “Football” example. The model consists of four levels, Activity, Event, Motion, and Object. The top three levels are mainly used by the query language processor to reason what activity the user wants to retrieve, so that the processor would retrieve the video scene from the database according to the objects (i.e., features with spatio-temporal semantics) from the bottom level of the model.

A user may input an activity into the Specification Language Processing component (shown in Figure 3.1 (c.1)) by using terms such as “playing sports”, “playing football”, “kicking football”, or “kicking”. More specific terms would yield more specific retrieval results. In general, the first word is known as a verb and the word followed is a noun. The processor would analyze from the Motion level first. After some keywords are matched, the processor would search up to the Event level using the second word. For example, if the term is “kicking football”, the processor searches “kicking” from the Motion level, and then uses “football” to search from the Event level. If the term is “playing football” and there is no “playing” in the Motion level, the processor will try to reason the thesaurus of the word and then search again. However, if there is no match of word from the model, the processor would just skip the “verb” and search the “noun” from the Event level to the Activity level. After the threshold of the search is met, the processor would go down to the corresponding Object level. Then it would input those objects from the Object level into the Feature Index.
Tree as features and ask the user to input some spatio-temporal semantics (ST-Feature) into the database (shown in Figure 3.1 (c.2)).

At a later time, the user may want to retrieve video data based on some activities from the database. For example, he may input an activity query like “kicking football”. The Query Language Processor first gets some collections of objects from the Activity Model (shown in Figure 3.1 (e)) and then retrieves the result as the original query processing (CAROL/ST) by treating the collections of objects as Features and ST-Features. Therefore, the main purposes of the Activity model are to facilitate annotating all common and significant activities.

### 3.2 CBR Extension to CAROL/ST

While CAROL/ST can facilitate effective retrieval based on rich semantics, for multimedia data such as video, visual content is also an inseparable (and can be more significant) part, which is difficult to be described with text. On the other hand, content-based approach to automatically extract and index visual features has been a main trend in the area of computer vision and video processing. To employ best strengths from both areas, an extended version of VideoMAP, which we termed as VideoMAP+ [CWLZ01], is developed for supporting hybrid retrieval of videos through both query-based and content-based accesses. Here we adopt visual content to our prototype only.

The architecture of VideoMAP+ is as shown in Figure 3.3 (which is a modified version of Figure 3.1). Here, the Feature Extraction Component (FEC) is newly added in. During the procedure of Video Segmentation (by VCC), visual feature vector of the video and other object defined in are extracted, such as the color, texture, shape and so on. The Hybrid Query Language Processing module contains three kinds of retrieval format: CAROL/ST Retrieval—the original retrieval format which mainly uses the semantic annotation and spatio-temporal relation of video. The Content-based Retrieval-module supports the newly added retrieval format that mainly uses the visual information inherent in the video content, and also their Hybrid Combination Retrieval. CBR query functions are incorporated to form a hybrid query language. Hence the indices are now based on more video objects and the returning result also includes more video object types.

**Figure 3.3 Architecture of VideoMAP+**
3.2.1 Foundation Classes

VideoMAP+ extends a conventional OODB to define video objects through a specific hierarchy (video→scene→segment→keyframe). In addition, it includes the concept of CBR to build index on visual features of these objects. Their class attributes, methods and corresponding relations form a complex network (or, a "map" as shown in Figure 3.4). Below we enumerate the foundation classes of the VideoMAP+ objects at various granularities, namely: Keyframe, Segment, Scene, Video and Visual object (cf. Figure 3.4).

![Figure 3.4 VideoMAP+ Foundation classes and their relationships](image)

As shown in Figure 3.4, the direct "bridge" between the two components (VideoMAP and CBR) of VideoMAP+ is at video segment level. This is not the only bridging level possible, as others (such as the keyframe and/or scene levels) are also meaningful for bridging the two. In VideoMAP+, the segment level is chosen as the direct bridge due to simplicity and efficiency reasons, because we regard video segments as the basic unit of retrieval.

3.2.2 Search paths with CBR

After integrating CBR with CAROL/ST, three main groups of objects (i.e. Keyframe, Visual-Object, and Image-Feature) are added to the VideoMAP+ system as shown in the class diagram (Figure 3.4).

**Image-Feature:** Visual Feature extracted from video object, like color, texture, shape and etc.

**Keyframe:** The fundamental image frame in video sequence.
**Visual Object.** All salient objects captured in a video’s physical space represented visually or textually are instances of a physical object. Furthermore, every object has the spatio-temporal layout in the image sequence.

Four new entry points to search for semantic-feature and visual-object are:

(a) Visual-Object,
(b) Image-Feature,
(c) Activity Model, and
(d) Object Level of the Activity Model.

The Object Level of the Activity Model [CL99b] contains annotated objects copied from the Activity Model, and it has links which link the semantic feature objects and the visual objects together.

**Text-based Search**

(1) There are two entry points to search for semantic feature. Entry point one is simply to search from the root of the semantic feature object collection. Another path is to search from the object level of the activity model, and then to retrieve the semantic features.

- Entry point 1: Semantic-Feature
- Entry point 2: (d) → Semantic-Feature

(2) To search for visual objects, there are also two entry points. Entry point one is simply to search from the root of the visual object collection. The other path is to search from the object level of the activity model, and then to retrieve the visual objects.

- Entry point 1: Visual-Object
- Entry point 2: (d) → Visual-Object

(3) It is possible for a user to search for activities occurred in a video program. Entry point one is to search from (c) to get the conceptual objects occurred in the activity. The conceptual objects are used as the semantic feature to find whether there are some video segments linked with the semantic features. Entry point two is to search from (c) to get the conceptual objects. The conceptual objects are used as the visual objects to find whether there are some video segments linked by the visual objects. Entry point three is to search from (c) to get the conceptual objects. Then these objects are used to search from (d) to obtain the related semantic features and visual objects. Since there are some direct links from (d) to the video segments, the last step is to search for the video segments linked by them.

- Entry point 1: (c) → Semantic-Feature → Semantic-Feature & Segment List → Semantic-Feature / Segment-List → Segment[Constraints: the semantic features are linked in the same video segments, and they all contain spatio-temporal features]
- Entry point 2: (c) → Visual-Object → Visual-Object List → Keyframe → Keyframe List → Segment
- Entry point 3: (c) → (d) → Segment List → Segment[Constraints: the semantic features and visual objects are linked together in the same video segments, and the semantic features contain spatio-temporal features]
Content-based Search

There are two paths to search for the low-level image features. Entry point one is simply to search from the image feature object collection. Entry point two is to search from (d) to get the visual objects. Then, using the link between the visual object and the image feature object, low-level image feature can be obtained.

- Entry point 1: Image-Feature
- Entry point 2: (d) \(\rightarrow\) Visual-Object \(\rightarrow\) Image-Feature

Hybrid Search

(1) To search for the low-level image feature of a semantic feature object, there are two search paths that can achieve. Entry point one is to search the image feature from the collection of image feature objects. Then, it is to get the visual object that is pointed by the image feature object. Using (d), semantic feature object can be retrieved. Entry point two is similar to entry point one but it starts from the end of the entry point one.

- Entry point 1: Image-Feature \(\rightarrow\) Visual-Object \(\rightarrow\) (d) \(\rightarrow\) Semantic-Feature
- Entry point 2: Semantic-Feature \(\rightarrow\) (d) \(\rightarrow\) Visual-Object \(\rightarrow\) Image-Feature

(2) To search for the low-level image feature of a visual object, the search paths are shown as follows.

- Entry 1. Image-Feature \(\rightarrow\) Visual-Object
- Entry 2. Visual-Object \(\rightarrow\) Image-Feature

(3) To search for activities occurred in a video program, it is first to use the low-level image features to search for the semantic feature objects and the visual objects (refer to the above point (1) and (2) of the Hybrid Search). Then follow the Entry point: Semantic-Feature & Segment List \(\rightarrow\) Semantic-Feature / Segment-List \(\rightarrow\) Segment.

3.2.3 Additional retrieval method

With respect to CBR, some additional video object comparison operators are defined and introduced into our query language, as summarized below.

(1) Keyframe similar_to Keyframe
   By color, shape, texture, or any combination;
   By visual objects (i.e. the number of similar visual objects).

(2) Segment similar_to Segment
   By the number of Keyframes deemed as similar (e.g. \(>50\%\));
   By the temporal ordering of Keyframes.

(3) Scene similar_to Scene
   By the number of Segments;
   By the temporal ordering of Segments;
   By the number of Activity/Event/Motion/Object from the Activity Model \[CL99b\].

(4) Video similar_to Video
   By the number of Scenes;
   By the temporal ordering of Scenes;
Those methods are based on a content-based video similarity model, whose implementation details are given in and can be referred to [WZ00b].

3.3 Language Syntax and Query Refinement

3.3.1 Syntax of CAROL/ST with CBR

The complete syntax of VideoMAP’s query language (i.e. CAROL/ST with CBR) is summarized below, with the shaded clauses being the ones not yet supported in the current version, but planned to be devised:

```sql
SELECT <object | attribute of object> [{,<object | attribute of object>}] 
FROM <object class> [{,<object class>}] 
[ WHERE <search condition> 
[{{AND <search condition>}}] 
[GROUP BY <object | attribute of an object> [HAVING <search condition>]] 
[ORDER BY <object | attribute of an object> [ASC | DESC] ] ;
```

For the WHERE clause, there are four kinds of search conditions:

1. Composite condition:
   `<object variable 1> HAS <object class> <object variable 2> [BY OID=<object id> | BY ONAME=<object name>]`
   If object variable 1 is linked with another object (either through the inheritance or composite relationships), object variable 1 can be retrieved by the composite condition.

2. Comparative condition:
   `<attribute of object> <comparison operator> <value>`
   If an object contains an attribute that is a simple data type, the attribute can be compared with the same simple data type.

3. Spatio-Temporal condition:
   `<object variable> AT <location operator> [FOR <comparison operator> <frame number>]`
   If an object is annotated with some spatio-temporal features, it can be retrieved by specifying its location and time duration.

4. Similarity condition:
   `<object variable 1> SIMILAR_TO <object class> <object variable 2> [BY OID=<object id> | BY ONAME=<object name>] [BY [COLOR | TEXTURE]]`
   If an object is associated with some visual feature objects, the object can be processed with the similarity measurement of images. Therefore, the <object class> above should be restricted to Keyframe, Segment, Scene, and Video objects.
3.3.2 Query refinement and feedback

Supported by CAROL/ST with CBR, queries in VideoMAP+ are flexible and able to accommodate various requirements. In particular,

- a single feature query will be simple to handle, since it uses the separated search schema: text-based retrieval or content-based retrieval. Text-based retrieval requires exact match; however, content-based retrieval adopts inexact ("fuzzy") match.

- a range query is a query that explicitly specifies a range of values for the feature weight, for example: 50% color, 30% texture, 20% shape. The weight for feature vector can be specified too.

- a heterogeneous feature query such as “find video segments similar to the sample segment1, and in addition satisfy the annotation restrictions” may involve convolution. Since text-based and content-based are both considered, different search paths and match methods should be incorporated. This problem can be tackled as follows:
  1) Intersection operation: The query can be any video object, maybe Keyframe, Segment, Scene, Video or Feature. Text-based semantic description can be used to narrow down the scopes of search. Only video object containing the specified semantic or spatio-temporal feature will be considered in the retrieving process.
  2) Join operation: Given a hybrid query, different sets of video object can be selected based on individual feature. The final ranked set of similar video objects to the query is derived by join operation.

Since Content-based retrieval is essentially fuzzy retrieval, it involves the problem of setting similarity threshold; k-means classification algorithm is used to dynamically find similar results, so in most situation, we get a set of similar video objects, not just a single one. While not all the video objects in the result set can fit for user’s requirement, or maybe new query ideas from the user can emerge at any time, further adjustment like query refinement and/or feedback is needed in VideoMAP+. Several ways are thus provided in order to support such query refinement.

1) Users can assign arbitrary video object from the result set as the starting point for a new round of query iteration.

   Under most situations, the query example used in the first round of iteration is not the best model for what the user actually wants to find, maybe it just comes from a coarse concept (a vague idea) that the user originally has. But after the result set gets returned, the user will be more clearly about what s/he wants and be able to pick a closer query example as the starting point for new query iteration.

2) Users can give feedback advices like “Relevant”, “No opinion”, “No Relevant” for the video objects in the result set.

   By feedback advice, user’s preference and intention on different features for the query can be analyzed. For example, if s/he prefers the red color more, then more weight on red color can be given so as to find more red objects. Feature weights can be adjusted in this way, and new query starts again. The feature weight adjustment algorithm we have been using is described in [WZ00b].
3.4 Summary

So far, we have described a hybrid approach to video retrieval in a comprehensive object-oriented video database system, based on the spatio-temporal semantics and visual features of video data. In addition, a query language supporting heterogeneous query (spatio-temporal reasoning query, semantic query and content-based query), namely, CAROL/ST with CBR, has also been introduced. It not only can overcome the drawback of CBR for the limitation of current computer vision and image processing techniques in extracting high-level (such as motion) semantics, but also bypasses the difficulty of describing visual content purely by text.

4. An Experimental Prototype

As part of our research, we have been building an experimental prototype system on the PC platform. In this section, we briefly describe our prototype work in terms of the implementation environment, the actual user interface and language facilities, and sample results.

4.1 Basic Implementation Environment & Functionalities

Starting from the first prototype of VideoMAP, our experimental prototyping has been conducted based on Microsoft Visual C++ (as the host programming language), and NeoAccess OODB toolkit (as the underlying database engine). Here we briefly introduce the main user facilities supported by VideoMAP*, the kernel system based on which our subsequent (web-based) prototype [LCWZ01] is being developed (using IONA’s developer-focused CORBA product, ORBacus, with necessary extensions). VideoMAP currently runs on Windows and it offers a user-friendly graphical user interface supporting two main kinds of activities: Video editing, and Video retrieval.

4.1.1 Video Editing

When a user invokes the function of video editing, a screen comes up for uploading a new video into the system and annotating its semantics (cf. Figure 4.1). For example, s/he can name the video object and assign some basic descriptions to get started. A sub-module of Video segmentation is devised to help decompose the whole video stream into segments and to identify keyframes. Further, the Feature Extraction module is to calculate the visual features of the media object. By reviewing the video abstraction structure composed by the segments and keyframes, the user can annotate the semantics according to his/her understanding and preference (see Figure 4.1). As a result, a video object is created and linked with a number of video segment objects. Each video segment objects contained keyframe objects and the low-level image features.

After creating and importing several video objects into the database, the user can create scene objects in a tree-like structure, and semantic feature objects in another tree. The user can also create attribute and method objects associated with the scenes and semantic features. A scene object (e.g. White_House_News) may contain a ToC object that can link up some video segment objects defined in several video objects dynamically. Semantic feature objects (e.g. Clinton, Lewinsky) can act as indexes which are attached to the
video segment objects in the ToC of a scene object dynamically. (For instance, Clinton may be attached to several different video segment objects in the ToCs of different scene objects; whereas Lewinsky may be attached to the others.) Users can specify a query to retrieve all scene objects of different video sources, by imposing the condition that the scene objects should contain both the semantic feature objects.

An activity hierarchy is created in the Specification Language component (see Figure 4.2) in order to facilitate the creation of visual objects and query retrieval. The Specification Language model can be used to annotate the spatio-temporal information to the semantic feature objects (cf. Figure 4.3), with a visual object being created and attached to the activity hierarchy meanwhile.

Figure 4.1 Annotating the segments after video segmentation

Figure 4.2 Creating an Activity Hierarchy in the Specification Language dialog
4.1.2 Video Retrieval

VideoMAP also provides an interface for the user to issue queries using its query language (i.e. CAROL/ST with CBR). All kinds of video objects such as “Scene”, “Segment”, “Keyframe”, can be retrieved by specifying their semantics or visual information. Figure 4.4 shows a sample query issued by the user, which is validated by a compiler sub-module before execution. The video objects retrieved are then returned to the user in the form of a tree (cf. Figure 4.4), whose node not only can be played out, but also can be used subsequently for formulating new queries in an iterative manner.
4.2 Some Results

4.2.1 On Query Types

As described earlier, VideoMAP supports three primary types of queries, namely: (1) query by semantic information, (2) query by visual information, and (3) query by both semantic and visual information (the "hybrid" type). VideoMAP supports retrieving all kinds of video objects (Video, Scene, Segment, Keyframe, Feature) based on the semantic annotations.

Query by Semantic Information

Figure 4.5 (a) and 4.5 (b) show the interface for users to specify a semantic query, and for displaying the result video, respectively. The semantic query of Figure 4.5 (a) uses an SQL-like dialect which is read as "select x from scene y where y has feature a [with oname="football"];". As shown in the Query Output panel of Figure 4.5(a), it returns three Scene objects that match the requirement.
Figure 4.5 (b) Displaying video result of the semantic query of Figure 4.5 (a)

**Query by Visual Information**

Users can also specify queries which involve visual features and their similarity measurement. Visual similarity considers the feature of color, texture and so on. Users can specify a query by using either individual features or their combination. Figure 4.5 (c) illustrates the interaction sessions of specifying a visual feature based query. Again, the visual query adopts an SQL-like syntax and is read as "select x from scene y where y similar_to scene a [with oname="football_game"] [by segment_num_color :> 50];". It returns four Scene objects that match the requirement. All scene objects are compared with the scene object "football_game" by using an algorithm of "percentage of segment by similarity measure of color" (cf. section 3.2.3 (3)). Due to the similarity measurement function, the results of this query type are usually of low precision but high recall.

**Query by Hybrid Information**

Users of VideoMAP can further submit queries which involve both semantic and visual features of the targeted video objects. These "heterogeneous" queries need different similarity measurements. A hybrid query example is given in Figure 4.5 (d) which reads as "select x from scene y where y similar_to scene a [with oname="football_game"] [by segment_num_color :> 50] and y has feature b [with oname="football"];". It returns two Scene objects which link with video segments and match the semantic information. However, the user may need a more precise result.; For example, he/she may launch a subsequent hybrid query as "select x from scene y where y similar_to scene a [with oname="football_game"] [by segment_num_color :> 50] and y has feature b [with oname="football"] and y.author = "Peter";", which is actually a more specific query than the previous one (see Figure 4.5 (e)).
Figure 4.5 (c) Query by visual information

Figure 4.5 (d) Query by hybrid search 1
**Query by Spatio-Temporal Information**

Figure 4.5 (f) and 4.5 (g) show the interface for users to specify a spatio-temporal query, and for displaying the video query result, respectively. The query of Figure 4.5 (f) can be read as "select x from scene y where y has feature a [with oname="football"] and a at middle_right for >= 2;". As shown in the Query Output panel of Figure 4.5(f), it returns two Scene objects that match the requirement.
Figure 4.5 (g) Displaying video result of the spatio-temporal query

4.2.2 On Query Refinement and Relevance Feedback

Query refinement and relevance feedback are important to improve system performance and provide the users with more flexibility in querying the database.

**Query Refinement**

From the current result set, a user can choose the most interested video object, and start a new iteration of query. For instance, Figure 4.6 (a) contains the result of an initial query by visual similarity, whose video object content is displayed in Figure 4.6 (b). Assuming the user finds a more interesting video object “movie1_segment7” which is close to what s/he wants to search, s/he can then specify it as the starting point for a new query iteration (cf. Figure 4.6 (c)). The result of this new query can thus be more precise, as shown in Figure 4.6 (d).
Figure 4.6 (a) Query by visual information

Figure 4.6 (b) Displaying result of the visual query
Figure 4.6 (c) Starting a new query iteration

Figure 4.6 (d) New query result by color information

Relevance Feedback:
Our prototype also allows users to specify from the result set the videos that are relevant, irrelevant, and/or indifferent, thereby assisting the system to conduct relevance feedback process as appropriate. In particular, assuming based on the result of Figure 4.7(a), the user can input a feedback advice of “cartoon_segment2” as “Not Relevant” as demonstrated in Figure 4.7 (b). For an instance, the user may not be interested in the
close-up segments of the monster. Through this feedback, “cartoon_segment2” is excluded by the new query, and there are only two segments remaining in the result set (cf. Figure 4.7 (a) and 4.7 (c)).

Figure 4.7 (a) Query by visual information

Figure 4.7 (b) Displaying the query result and marking a segment as irrelevant
5. Conclusion and Future Research

To adequately accommodate the ever-increasing demand for video retrieval over the web, we have developed a comprehensive video data management system, viz., VideoMAP. In this paper we have presented its central mechanisms which offer a number of unique features. Among others, it provides a hybrid approach to video retrieval, by combining query-based method with content-based retrieval (CBR) functions. We have also described the user interface facilities of our system, particularly query specifications in CAROL/ST with CBR; other facilities such as query refinement and relevance feedback are also devised and presented from the perspective of sample user interactions.

There are a number of issues for us to further work on. Besides building up a web-based version on a web server, we are extending our prototype system to make it MPEG standard compatible, by allowing input videos to be in either AVI or MPEG format. The cost of data transfer and communication is also an important issue worth great attention. In order to reduce this overhead, databases can be fragmented and distributed into different sites. There are many research efforts on data fragmentation of relational databases. On the contrary, there is only limited recent work on data (object) fragmentation in object databases [KL00, ODV94, OV99]. The next important issue is the fragment allocation, which is another open research issue for object databases [SAK99]. We also start to examine the possible performance issues, such as replication technique. Finally, the issues of optimizing CAROL/ST with CBR queries, and the possibility of combining local query processing with global query processing in the context of a distributed, web-based VideoMAP are interesting for further investigation.
Acknowledgement - This work has been supported, in part, by City University of Hong Kong under her Strategic Research Grant (no. 7001073), and by the Research Grants Council of the Hong Kong SAR, China (project no. CityU1119/99E).

References


[HARMONY] The Harmony Project <http://www.lirt.bris.ac.uk/discovery/harmony/>


